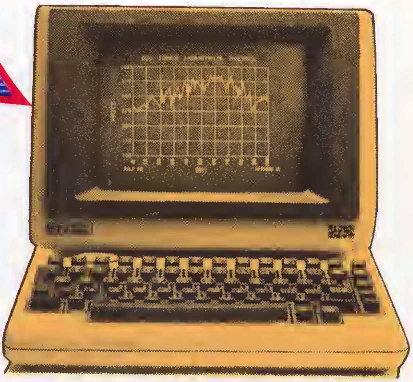
A BYTE BOOK

>>$9.95

HOW TO MAKE YOUR COMPUTER TALK

STEVEN J. VELTRI

For: Apple II, IIplus, and IIe; TRS-80 I, III, and IV; Commodore 64 and VIC 20; Sinclair ZX80 and ZX81; and Timex/Sinclair 1000



HOW TO MAKE YOUR COMPUTER TALK

Steven J. Veltri

McGraw-Hill Book Company

New York St. Louis San Francisco
Auckland Bogotá Guatemala Hamburg
Johannesburg Lisbon London Madrid
Mexico Montreal New Delhi Panama
Paris San Juan São Paulo Singapore
Sydney Tokyo Toronto

Radio Shack and TRS-80 are trademarks of Tandy Corporation.
Apple II, IIplus, and IIe are trademarks of Apple Corporation.
Sinclair ZX80 and ZX81 are trademarks of Sinclair Research Limited.
T/S 1000 is a trademark of Timex Corporation.
VIC 20 and Commodore 64 are trademarks of Commodore, Inc.
SPO256-AL2 is a trademark of General Instrument Corporation.

The author of the programs provided with this book has carefully reviewed them to ensure their performance in accordance with the specifications described in the book. Neither the author nor McGraw-Hill, however, makes any warranties whatever concerning the programs. They assume no liability or responsibility of any kind for errors in the programs or for the consequences of any such errors.

HOW TO MAKE YOUR COMPUTER TALK

A BYTE Book

1 2 3 4 5 6 7 8 9 0 KPKP 8 7 6 5 4

ISBN 0-07-067380-2

Library of Congress Cataloging in Publication Data

Veltri, Steven J.
How to make your computer talk.

(A Byte book)
Includes index.
1. Speech synthesis. I. Title. II. Series: Byte books.
TK7882.S65V45 1985 001.64'43 84–11281
ISBN 0-07-067380-2

The editor for this book was John Aliano.
The editing supervisor was Margery Luhrs.

Book Design by Patrice Fodero.

CONTENTS

PREFACE

Why is there a growing interest in adapting speech to the home and personal computers?

Talking. Defined in Webster's dictionary as the power to *communicate* to an individual or group of individuals. This ability to put ideas into words has become the most natural form of human communication in the world today.

Computers. Defined in Webster's dictionary as an *electronic* machine that performs rapid calculations and compiles or correlates *information.*

Talking Computers. Combining the two, therefore, offers the advantage of *communicating information electronically* to an individual or group of individuals via the spoken word.

The computer that can communicate with its owner more naturally has a certain personality and is easier to use. A talking computer is just one step closer to becoming a "friendly" computer.

Until recently, talking computers existed only in the fantasy of science fiction or at a great cost to industrial and university laboratories. Since then, speech synthesis units have appeared commercially for $250–$350. This book explains how you can build your own speech synthesis unit for the Apple II, IIplus, IIe; TRS-80, Models I, III, IV; Commodore 64, VIC 20; Sinclair ZX80, ZX81, or Timex/Sinclair 1000 for under **$35.**

In addition to the tremendous cost savings, this book offers an introduction to speech synthesis theory and will help you under-

PREFACE

Why is there a growing interest in adapting speech to the home and personal computers?

Talking. Defined in Webster's dictionary as the power to *communicate* to an individual or group of individuals. This ability to put ideas into words has become the most natural form of human communication in the world today.

Computers. Defined in Webster's dictionary as an *electronic* machine that performs rapid calculations and compiles or correlates *information.*

Talking Computers. Combining the two, therefore, offers the advantage of *communicating information electronically* to an individual or group of individuals via the spoken word.

The computer that can communicate with its owner more naturally has a certain personality and is easier to use. A talking computer is just one step closer to becoming a "friendly" computer.

Until recently, talking computers existed only in the fantasy of science fiction or at a great cost to industrial and university laboratories. Since then, speech synthesis units have appeared commercially for $250–$350. This book explains how you can build your own speech synthesis unit for the Apple II, IIplus, IIe; TRS-80, Models I, III, IV; Commodore 64, VIC 20; Sinclair ZX80, ZX81, or Timex/Sinclair 1000 for under **$35.**

In addition to the tremendous cost savings, this book offers an introduction to speech synthesis theory and will help you under-

stand the basic speech concepts. The applications described will show you how speech synthesis can be a useful tool for your programming needs.

Many books have been written about speech synthesis theory; however, this is the first of its kind to simplify a complex technology so that it's understandable to any computer owner. It is written in clear, concise layman's terms. It not only offers a step-by-step procedure for building and operating a speech unit for your home computer, but also educates you in this exciting new technology. It's true, there are other speech synthesis products and books on the market, but none of them offers the combined advantages just stated. So why invest large sums of money in other speech synthesis systems when this book can teach you how to build your own for less than the amount of money you might spend in a restaurant? After reading this book, *you* will have the ability to make the final decision: Do I want a speech synthesizer for my computer? If this book convinces you to explore this technology further, refer to the Parts Supplier Listing in Appendix B. If, however, you decide against purchasing a speech unit, you've still saved money by being convinced that a speech synthesis product is not what you want. In addition, you will retain a basic knowledge of this new technological breakthrough. A small investment for such a vast wealth of knowledge!

As stated above, this book was written to be both educational and informative. It describes how the human speech mechanism works, and explains the simple physical model that can be used to represent it; it is interesting to see the similarities between the two, which, of course, enable computers to talk. And because this book is an introduction to speech synthesis using home computers, it includes software examples that illustrate the fundamental principles and techniques. In addition, there are some sample programs which enable you to add speech to your own programs.

This book is divided into ten chapters that present speech synthesis concepts, followed by fully tested hardware circuits and software programs.

CAUTION: Please note it is extremely important that all cautions and references to safety cited in this book are strictly

adhered to. Further, the author has taken every precaution to ensure that all procedures are safe and will not present any danger to the user. However, any electrical procedure can be dangerous and neither the author nor McGraw-Hill makes any warranties whatever concerning the procedures described. They assume no liability or responsibility whatever for possible injuries that may result from improper handling of the equipment and parts.

Chapter 1 discusses how the human vocal tract constructs sounds that we interpret as speech. It explains how speech begins at the vocal cords as sounds, and how these sounds are transformed into speech. By understanding the basic theory of speech production one can understand how computers can be programmed to talk.

Chapter 2 describes speech synthesis techniques in common use today. It also explains some of the advantages and disadvantages of each technique as related to the home computer.

Chapter 3 discusses the speech synthesis technique chosen to give your computer an unlimited vocabulary. It explains in detail how sounds are classified and how these classifications are used when synthesizing speech. This chapter also explains, using examples, how your synthesizer can make your computer talk.

Chapters Four, Five, Six, Seven, and Eight describe the speech hardware interfaces to the Apple, TRS-80, Commodore, Sinclair, and Timex/Sinclair computers. Sample programs are given both to test the hardware and to illustrate the fundamental computer-generated-speech principles. Directions that explain how to add speech to your own software programs are included. These chapters describe how to build synthesizers for *any* computer-owner with minimum to advanced experience. Each chapter describes the construction of starter kits for the advanced hobbyist or complete kits for the beginner hobbyist, and details the complete module for the person who doesn't have the time or desire to build one himself. Program listings, schematics, parts lists, and operating instructions are all included.

Chapter Nine describes the operation of the speech synthesizer chip chosen. It explains the chip's interfacing requirements, timing requirements, and all its pin functions and electrical characteristics. This chapter is intended for the *advanced* hobbyist who wants to experiment with the synthesizer chip on his or her own. With the information supplied in this chapter, an *experienced* designer can interface this device to any 8-bit microprocessor or microcomputer.

Chapter Ten introduces you to an area known as Text-to-Speech. Text-to-Speech is an "add-on" software program that requires the speech synthesizer constructed in Chapters Four, Five, Six, Seven, and Eight. This chapter explains what Text-to-Speech is, why it is useful in certain types of speech applications, and explains its advantages and disadvantages.

Appendix A contains a dictionary of over 250 of the most commonly used words. The dictionary lists each word and the sounds used to create that word. The dictionary enables you to easily build words and phrases for your speech synthesizer.

Appendix B offers a list of suppliers for the parts you'll need to build a speech synthesizer, or where you can purchase a complete module for your particular computer.

Glossary of Terms. At the beginning of Chapters One, Two, and Three is a list of technical terms and definitions used in that chapter. Written in this fashion, each chapter becomes easier to understand because you are sure to be familiar with key words encountered while reading the chapter.

Let's Get Technical. At the end of each chapter are sections that contain further technical information. These sections are intended for readers with some background in electronic circuit design and speech synthesis theory.

The book is formatted in such a way to be useful for the beginner hobbyist (with little or no electronic speech background), the advanced hobbyist or designer, the student, or *any* computer-owner interested in giving his or her computer the power of speech. It can serve as an introduction to speech synthesis for the beginner,

as well as a technical reference for the advanced hobbyist. No previous experience in speech synthesis or circuit assembly is required.

After reading this book, you will have taken the first step to understanding the basic concepts of speech synthesis and its use with home computers. It gives you the ability to make your computer into a "talking computer." By utilizing the information explained in this book, you will be able to experiment with speech applications on your own. A few suggestions are:

Game Application Programs. Fierce competition between you and your computer can now be guided and accentuated with speech. Have your space games tell you when you're running low on fuel; keep your eyes on the game instead of the clock; have the synthesizer count down your time. Inevitably, your game scores will be improved.

Educational Programs. Generate a program for your child that will hold his or her interest through listening and learning. For instance, separating the sounds in large and small words will prove to be a vital asset in their learning experience. Adding speech in this ever demanding world of visual aids opens a new avenue to the educational applications of your computer.

Household Applications. Relieve the pressure of "clock-watching." Program your computer to give you a call when you're on a tight schedule and your time in the shower is running out; call you when your roast is ready; or tell you when your favorite TV show is about to come on—the possibilities are endless in solving everyday problems. For your home security system, have the synthesizer announce the location of a disturbance and the safest exit from the house. Use your computer as a telephone answering machine and have the synthesizer inform the caller that you are unable to answer the phone and to please leave a message.

Handicapped Applications. Your computer can also provide speech for the handicapped. This, in my opinion, is one of the major assets of this new technology.

As stated, this book gives you the tools you need to create speech applications on your own. You are limited only by your imagination. The power of synthesized speech is now witnin your reach.

Steven J. Veltri

ACKNOWLEDGMENTS

I'd like to thank my following friends and colleagues who have helped me with the writing of this book in various ways. Richard Giglia, president, Research In Speech Technology, Inc.; Rebecca Moore, technical consultant; Edward Garrity, technical consultant; Dennis Intravia, technical consultant; Jeanne McLaughlin, administrative supervisor, Brian Hopkins, art designer; Virginia Hopkins, administrative assistant; Dorsey Harrington, literary consultant; and Maria Cataldo, assistant V.P., Research In Speech Technology. Their help and support has made this project possible.

CHAPTER ONE

THEORY OF SPEECH SYNTHESIS

GLOSSARY

ARTICULATION The process of positioning or connecting the articulators and changing the shape of the vocal tract when producing speech.

ARTICULATOR A medium by which sounds are created. In the human vocal tract the articulators are the lips, tongue, palate, and teeth, and they shape the human oral cavity.

FORMANTS The resonant frequencies of the vocal tract.

PITCH The highness or lowness of a sound due to vibrations of sound waves.

RESONANT FREQUENCY A sound frequency that is intensified or allowed to pass through a filter. The filter screens out or allows certain frequencies to pass through.

SOUND FREQUENCY Any particular sound is measured by the number of times the sound wave oscillates in a given period. A sound frequency is measured in cycles per second or *hertz* (Hz). For example, the sound frequency of human speech ranges from 0–5,000 cycles per second or 0–5,000 Hz.

TRACHEA The throat or windpipe.

UNVOICED SOUNDS Sounds that are produced in the vocal tract when the vocal cords are not vibrating.

VOCAL TRACT In humans, it mainly consists of the articulators, oral cavity, nasal cavity, and throat.

VOICED SOUNDS Sounds that are produced in the vocal tract when the vocal cords are vibrating.

THE MIRACLE OF SPEECH

In order to understand how computers are able to speak, it is important to understand how humans speak. Human beings produce speech in much the same way that flutes and horns produce their sounds. In the case of the horn, the player blows into one end of the horn with a kind of "humming" sound. The horn vibrates because the player's lips are vibrating. The physical characteristics of the horn (i.e., size, shape, length, etc.) determine the way the horn vibrates. This gives each kind of horn its own particular tone or sound. When different types of horns play the same musical note, or *pitch,* they don't sound alike. This is because some frequencies are absorbed while others are allowed to pass through, depending upon the shape and size of the horn. The frequencies that are absorbed are not heard. The frequencies that pass through are heard. This gives every horn its own characteristic sound. The frequencies that are passed through are known as *resonant frequencies.*

Human speech is produced in much the same way. The human vocal cords vibrate, like the lips of the horn player, and a variety of sound frequencies enter the throat. They then begin to pass up and out through the mouth. Just as the horn, the size and shape of the mouth determine which of these frequencies will be heard (passed through) and which ones will not be heard (absorbed). The miracle of speech is accomplished because the shape of the human throat and mouth can be changed at will (see Fig. 1-1), thus giving the speaker the ability to determine which frequencies will be heard and which ones won't. It is the sequence of these frequency selections that we interpret as speech.

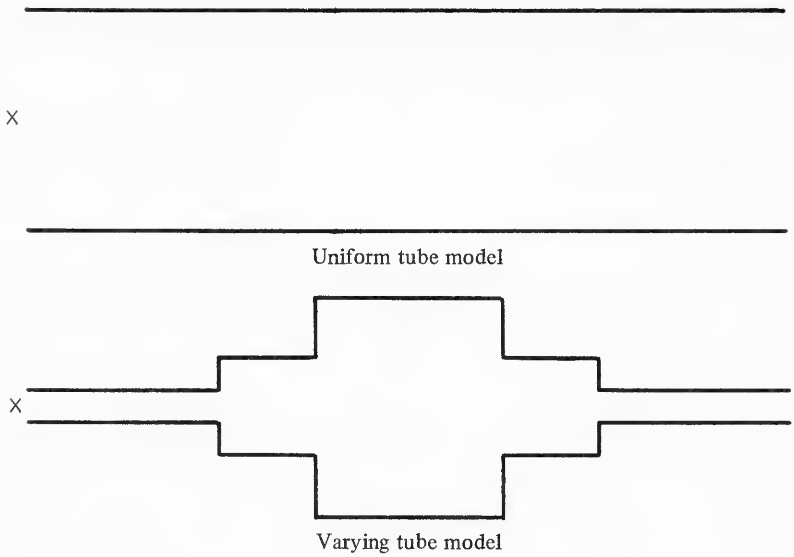
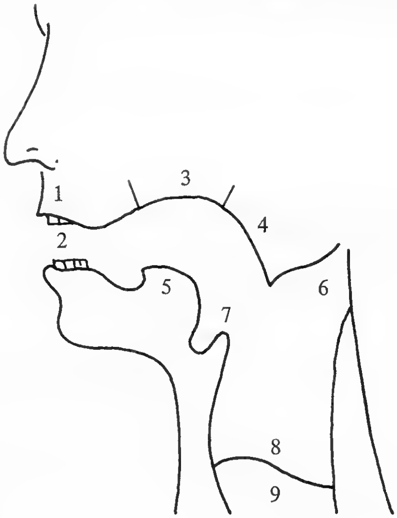


Fig. 1-1 A simplified model of the human vocal tract that illustrates the continuously varying shape of the vocal tract. × represents the vocal cords.

ARTICULATION

The human brain positions the articulators—the lips, teeth, palate, and tongue, along with the throat muscles in order to create the desired changes in size and shape (see Fig. 1-2). We associate any given word with a particular series of selected frequencies, and thus, with a particular series of positions of the human vocal tract. Even something as powerful as the human brain takes several years to master this control.

The speech system then can be considered as consisting of a series of tubes and cavities connecting the lungs to the mouth and nose. These tubes and cavities are about 7 inches long. The vocal cords, located at the opposite end of the trachea from the lungs, control the flow of air from the lungs to the vocal tract. Under muscular control, the cavities that make up the vocal tract can significantly change shape at a rate of 10 times per second, and



1. Lips
2. Teeth
3. Hard palate
4. Soft palate (velum)
5. Tongue
6. Pharynx
7. Epiglottis
8. Position of vocal chords
9. Glottis

Fig. 1-2 The organs of the speech system. At any instant in time only a few of these elements are in use, producing or influencing the sound. Placement of the articulators (lips, tongue, palate and teeth) determines which frequencies are absorbed by the vocal tract and which frequencies are allowed to pass through the vocal tract. Therefore, generating speech from the vocal cords involves varying the shape of these cavities.

the vocal cords can open and close at a rate of approximately 100–300 times per second. The changing shape of the vocal tract, and the shape and positioning of the articulators is known as *articulation.*

Voiced versus Unvoiced Sounds

When whispering, human beings produce recognizable speech without moving their vocal cords. This should be further evidence to the reader that it is the movement of the vocal tract that is important and not the movement of the vocal cords. In fact, during normal speech human beings produce sounds with and without vibrating their vocal cords. These two types of sounds are classified as *voiced* (Fig. 1-3) when the vocal cords are vibrating, and *unvoiced* (Fig. 1-4) when they are not vibrating. Examples of voiced sounds are the vowel sounds heard in the following words: **a**t, **ea**t, **i**t, **oa**t,

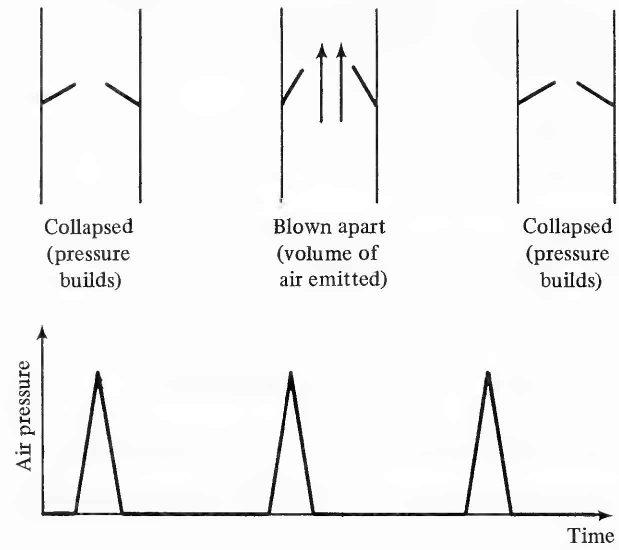


Fig. 1-3 A uniform tube model of the human vocal tract producing voiced sounds (*top*). This creates the energy source that results in pulses of air which pass through the vocal cavities (*bottom*). These pulses of air form the sound frequencies that are either passed through or absorbed by the vocal tract.

and flute, and some consonant sounds heard in words such as man and nine. To demonstrate the vocal cord vibration, place your hand on your throat while pronouncing these sounds. You can actually feel your vocal cords vibrating.

Examples of unvoiced sounds are heard in the words stop and

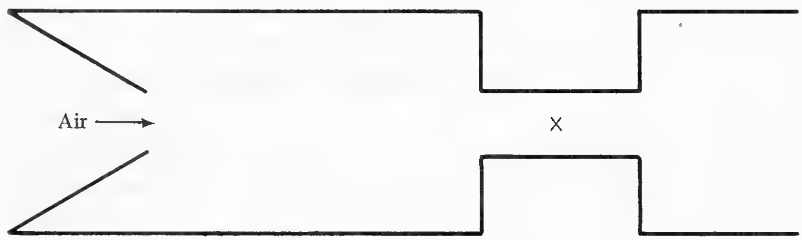


Fig. 1-4 A small opening in the vocal cavity formed by the teeth, tongue, or lips at the front of the vocal tract. Air is forced through this opening when forming unvoiced sounds.

sheet. If you place your hand on your throat once again and pronounce only the "s" or "sh" sound, you will notice that you will not feel any vibration. The sound is actually being produced up near the teeth and lips by forcing air through a small opening between the tongue, teeth, and lips. This is the same effect that produces a hiss when air rushes out of a tire under pressure.

KEY FEATURES FOR SPEECH

From the above discussion, we see that the three key features used in the human vocal tract to produce speech are:

1. Vibration of the vocal cords for voiced sounds
2. Forcing air through a small opening for unvoiced sounds
3. Change in size and shape of the vocal tract for frequency selection

Some of the methods of artificial speech production described in Chapter Two reproduce only the effects of the human vocal tract. Other methods actually imitate the operation of the three major features listed above.

LET'S GET TECHNICAL . . .

We have just learned that human speech consists of two classifications of sound frequencies. Voiced sounds, which are produced by vocal cord vibration, and unvoiced sounds, which are produced by forcing air through small constrictions in the vocal cavity when the vocal cords are not vibrating. Both of these sound frequencies enter the vocal tract and are passed through or absorbed by the vocal tract. The vocal tract is actually *filtering* out the undesirable sound frequencies. Therefore, the vocal cavity can be considered to be an *adaptive filter network*.

The varying shapes of the vocal cavity and placement of the articulators alter the resonances and frequency response of this

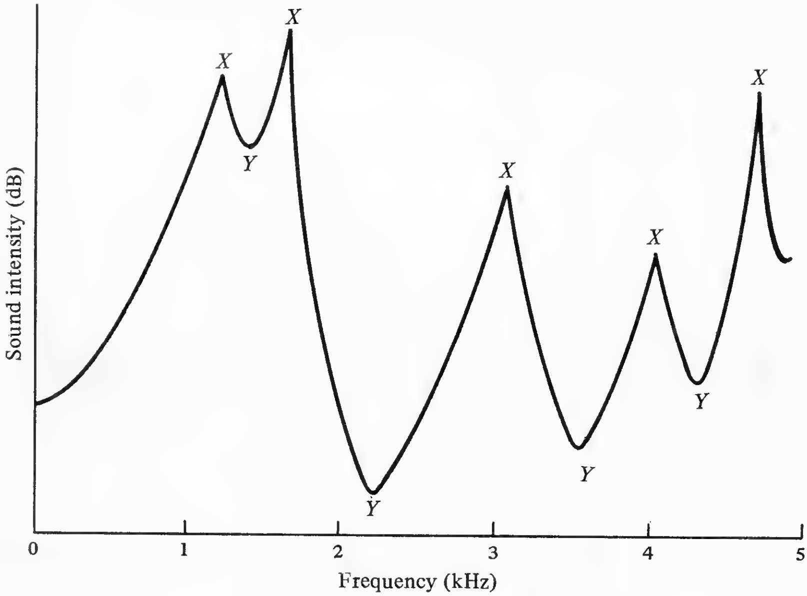


Fig. 1-5 The frequency response of a filter (e.g., horn, vocal tract, sound filter, etc.) at a point in time. If we put all frequencies from 0–5 kHz into the filter, at equal volumes, they would be absorbed or passed according to the graph. X represents the peak frequencies (formants) that are passed. Y represents the frequencies that are absorbed.

network. The resonances of this filter network are known as *formants*. These are the peaks in the frequency plot of the speech signal measured in decibels (dB), as seen in Fig. 1-5. In mathematical terms these are the poles of the transfer function of the filter network at any instant in time. So when the vocal tract network changes, the frequency response changes and the poles of the transfer function change (i.e., adaptive response). These poles are analyzed, and correlating values are applied in an electronic model of the vocal tract when speech is synthesized.

SUMMARY

Speech is produced by vocal cord vibration (voiced sounds) or constrictions in the vocal tract (unvoiced sounds). These sounds

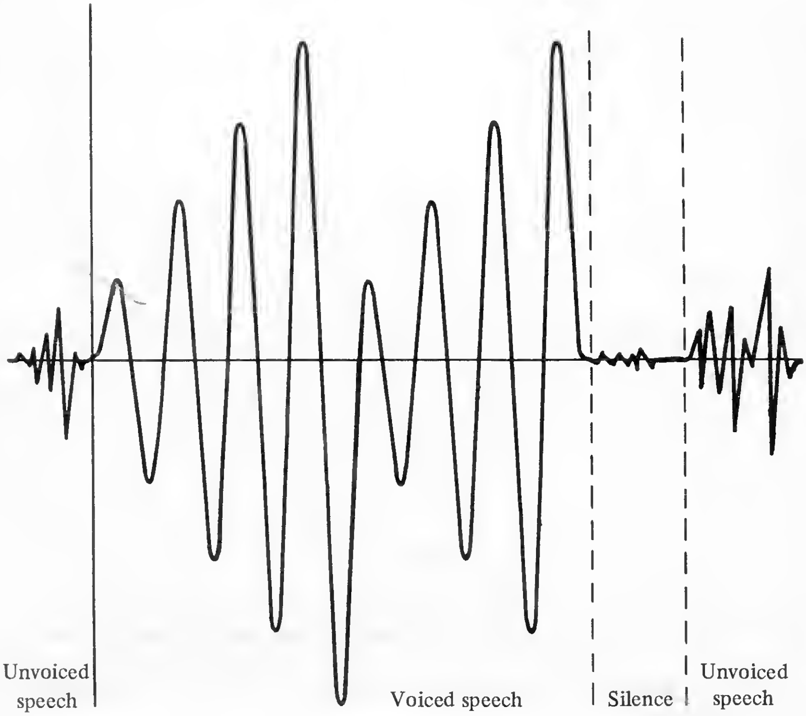


Fig. 1-6 A complex waveform produced by the vocal tract. The silent portion of speech is produced in the transition from voiced to unvoiced sounds. This signal is made up of many complex frequencies.

pass through the vocal tract which constantly changes shape, thereby changing the frequency response of the signal. The output signal after being filtered in the vocal cavity is known as speech. The speech signal can be regarded as a waveform consisting of many complex frequencies at any instant in time (see Fig. 1-6). The vocal tract can be thought of as a complex time variant filter network used to produce these complex speech frequencies.

CHAPTER TWO

SPEECH REPRODUCTION TECHNIQUES

GLOSSARY

AMPLITUDE In waveforms, the range from the zero axis (mean) to the maximum value (extreme). In sound waves the amplitude can be regarded as the volume of the signal.

ANALOG FILTER A device that is used to screen out or pass through analog waveforms.

ANALOG-TO-DIGITAL CONVERSION The process of converting an analog waveform to a digital waveform.

ANALOG WAVEFORM A waveform that is continuous.

DIGITAL-TO-ANALOG CONVERSION The process of converting a digital waveform to an analog waveform.

DIGITAL WAVEFORM A waveform that is used to represent an analog waveform using digital values (numerical codes in base two).

DISCRETE LOGIC Transistor transistor logic (TTL) which comprises electronic gates (e.g., NOR, AND, etc.).

DYNAMICS In human speech, the process of varying pitch in any utterance.

ENERGY SOURCE A device used to produce electronic pulses.

INTONATION The manner of producing tones with regard to accurate pitch.

PERIODIC SOURCE An energy source that produces electronic pulses at regular intervals or periods.

PSEUDORANDOM NOISE SOURCE An energy source that produces electronic pulses at random intervals.

SAMPLING FREQUENCY The rate at which an analog waveform is "looked at" when converting to a digital waveform.

Three methods of human speech reproduction and synthesis are discussed in this chapter. The first method is *pulse code modulation* (PCM) which is used to digitally record and playback speech waveforms. The second method is *continuously variable slope delta modulation* (CVSD) which is a cousin of PCM. The third method is called *linear predictive coding* (LPC) and uses a speech synthesis model to imitate the operation of the human vocal tract. All three methods begin with the same basic steps. One step incorporates an analog-to-digital conversion of audio signals. Another step incorporates a digital-to-analog conversion of audio signals. First let's review analog-to-digital conversion.

ANALOG-TO-DIGITAL CONVERSION

Analog-to-digital conversion (A/D) operates in the following way. Each point of the original waveform is represented by a number (digital code) which corresponds to the amplitude of the signal. The so called "sampling rate" determines how often the waveform is measured. For example, the temperature outside your house is constantly changing. If we decided to do an analog-to-digital conversion of the temperature, we might agree to measure the temperature at noon every day and round it off to the nearest degree. If we continued this for 30 days we would have a digital representa-

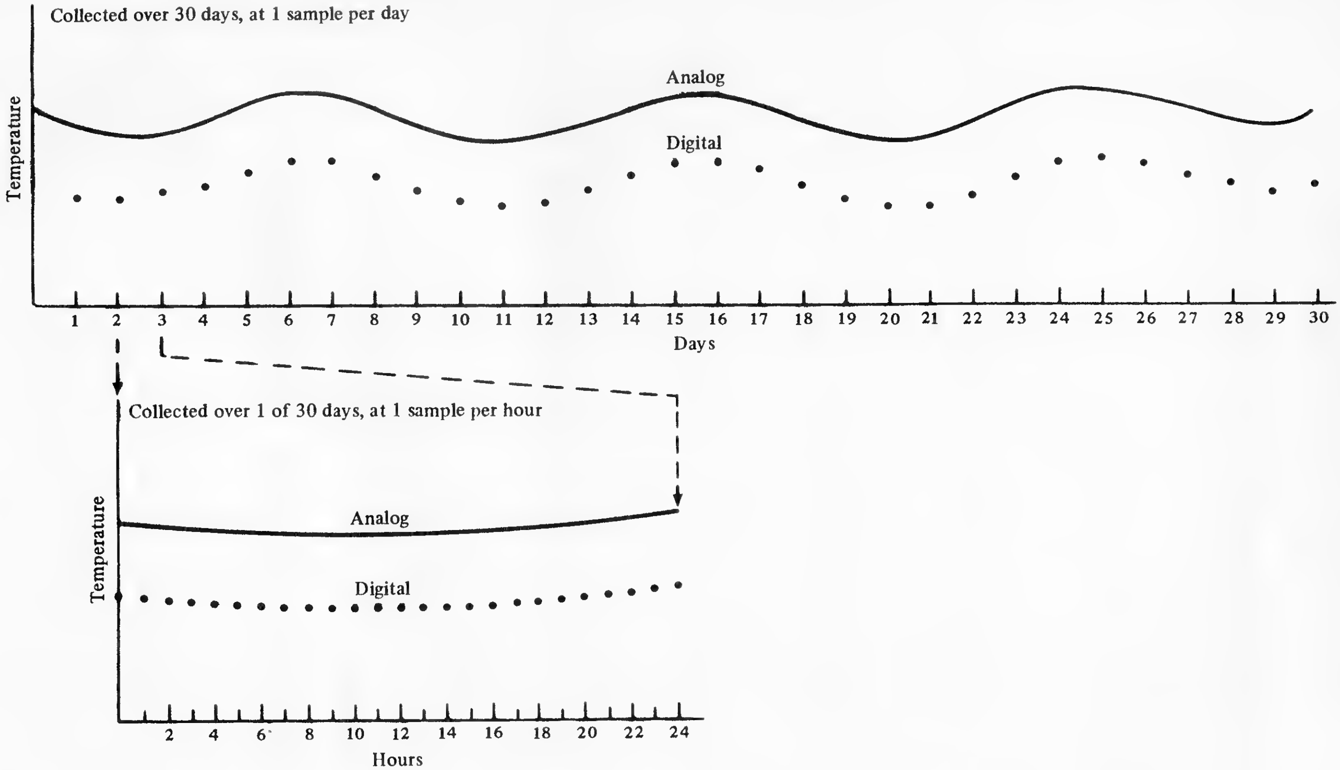


Fig. 2-1 An analog-to-digital conversion of a continuous waveform being sampled at the rate of one sample per day (*top*). An analog-to-digital conversion of the same waveform sampled at the rate of one sample per hour (*bottom*).

tion of the temperature outside your house. The sampling rate would be equal to one sample per day. The fact that we rounded the measurement off to the nearest degree means that although the temperature might have changed very slowly at a continuous rate, each of our measurements would indicate that it changed in steps of one degree. If we wish to get more information about temperature movements, we may take samples once per hour instead of once per day (see Fig. 2-1). So *analog-to-digital conversion* is characterized by representing an analog waveform by a series of numerical values spaced at equal intervals of time.

When synthesizing human speech, the first step is to make a high quality recording of the speech to be synthesized. Professional speakers (orators) produce the best overall results because their voices have pleasing intonation and dynamics. The recorded signal is passed through an analog filter to remove frequencies above one half the sampling frequency. The signal is then passed through an analog-to-digital converter at a desired sample rate. The sampling rate should be at least twice the highest frequency found in the analog signal for the frequency information to be accurately preserved. Reproducing frequencies in a speech signal up to 5 kHz will give a good reproduction of human voice quality. In order to accomplish this reproduction, we must filter the voice waveform to eliminate frequencies above 5 kHz and then digitally sample it at 10 kHz. Sampling this signal means looking at it 10,000 times in one second and recording the data. It is this data that is used for each of the three methods of human speech reproduction that will be described.

DIGITAL-TO-ANALOG CONVERSION

Digital-to-analog conversion (D/A) operates in much the same manner as analog-to-digital conversion; however, the process works in reverse. The digital codes in a digital waveform correspond to amplitudes in an analog waveform (see Fig. 2-2 *top*). For each digital code the amplitude level remains constant until there is a significant change in the digital code. When this change occurs,

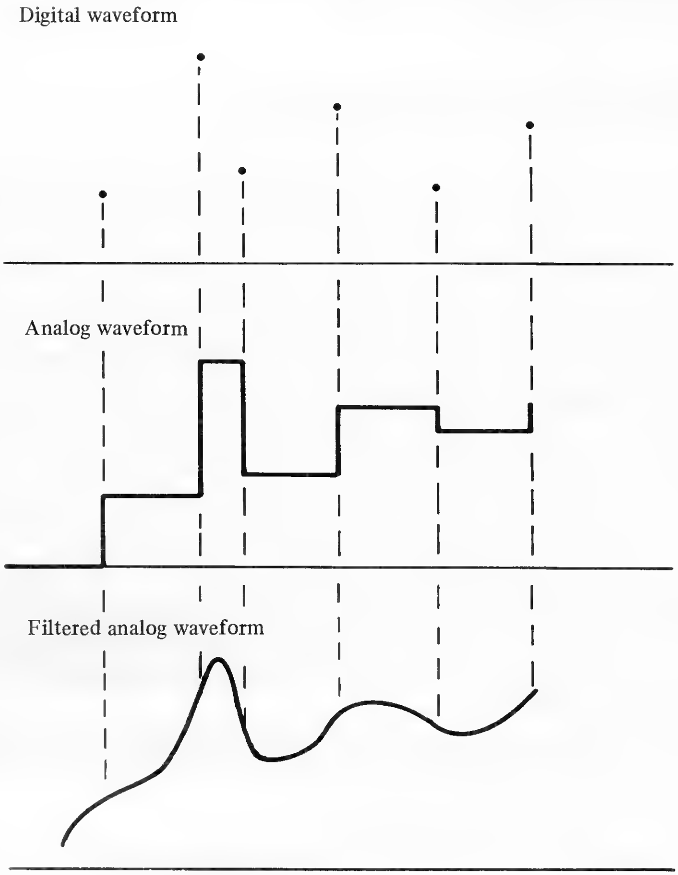


Fig. 2-2 Digital-to-analog conversion from an analog waveform.

the amplitude in the analog waveform also instantaneously changes (see Fig. 2-2 *middle*). This process produces a continuous analog waveform that is further filtered to produce the final analog waveform (see Fig. 2-2 *bottom*).

After synthesizing human speech, utilizing analog-to-digital conversion and one of the three methods described in the following sections, the digital data is stored in memory. This data must

be passed through a digital-to-analog converter and speaker for the synthesized speech to be heard.

PULSE CODE MODULATION

PCM offers the highest quality of speech reproduction, but unfortunately it also requires larger amounts of data storage than most home computers possess, even for small vocabularies. Simple additional hardware such as a digital-to-analog converter and a small amount of discrete logic will adequately perform PCM. Fairly high-quality speech can be reproduced at a memory requirement of 70,000 bits of memory storage per one second of speech. An average word will require about 35,000 bits of storage in this form, assuming a 0.5-second duration for the average spoken word. Since we are talking about adding speech to computers that contain approximately 128,000 bits of memory (16K bytes), you could store approximately four words in the entire computer's memory.

In PCM, each different amplitude of the speech signal is represented by a different digital code (see Fig. 2-3). No assumptions are made about the nature of the signal or its relationship with the speech mechanism.

A recorded or live speech waveform is passed through an ana-

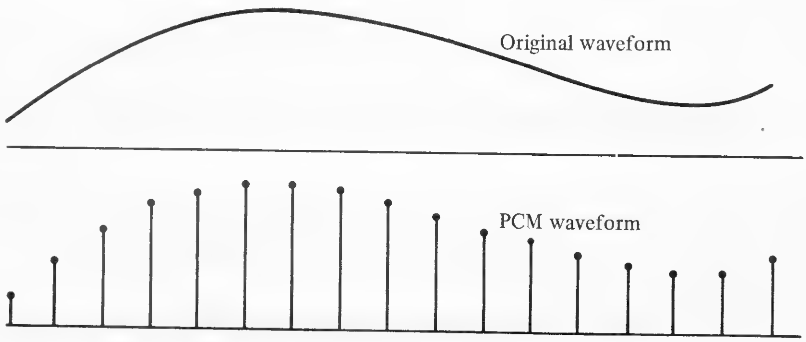


Fig. 2-3 Waveforms that are produced using PCM techniques. Each point represents the value of the original waveform at any instant of time. Each number (digital code) is called a sample.

log-to-digital converter as previously described and is stored in memory. The quality of the speech, after passing it back through a digital-to-analog converter, depends on several factors. The *sampling frequency* (or how many times each second the waveform is "looked at" and its value digitized) is the most crucial factor. The higher the sampling rate, the closer it will be to the original recording. The minimum value of the sampling frequency must be twice as high as the highest frequency in the original waveform being sampled (e.g., if the highest frequency in the original waveform is 5 kHz the minimum sampling frequency should be 10 kHz).

CONTINUOUSLY VARIABLE SLOPE DELTA MODULATION

Unlike PCM, which assigns a digital code to each amplitude, CVSD assigns a digital code to represent only the change in amplitudes of adjacent samples (see Fig. 2-4). CVSD will also vary the amount of change represented by a given code when it has been preceded

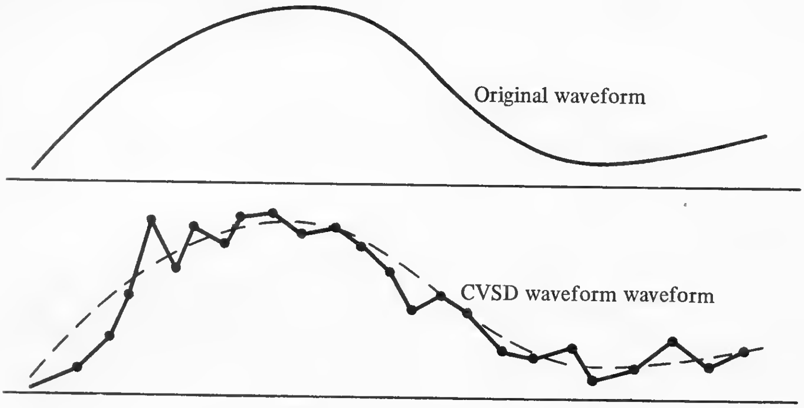


Fig. 2-4 Waveforms using CVSD techniques, which records a change in adjacent samples.

by certain other sequences of codes. CVSD speech synthesis requires less memory storage than the PCM technique discussed so far; generally 16,000 bits per second, or 8,000 bits per word. Using our same example as above, a 128K computer (16K bytes) would be able to store sixteen words.

LINEAR PREDICTIVE CODING

Using LPC, speech waveforms can be coded and stored using only 1,000–2,000 bits per second. (In some cases it may be as high as 5,000 bits per second.) Although the hardware used in LPC synthesis is fairly complex, the savings in memory storage more than offsets any additional cost in synthesis hardware. At this rate, we can store approximately 128 words on our 128K computer (16K bytes).

LPC capitalizes more specifically on *how* the speech waveform is produced rather than merely *reproducing* the waveform, as with the other two techniques. Its name was derived from the technique it uses to synthesize speech. It *predicts* the next *coded* speech sample by using a *linear* combination of the preceding speech samples. It actually imitates the features reviewed in Chapter One.

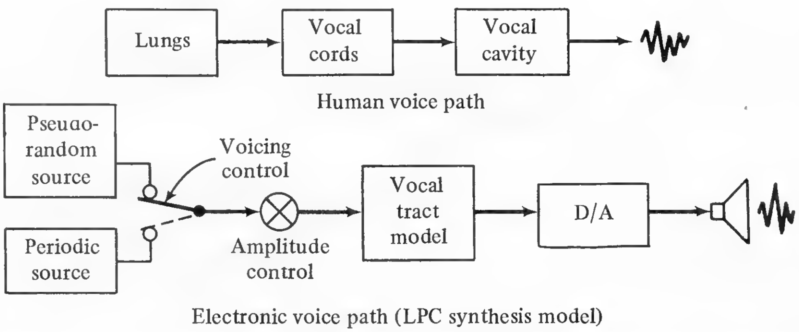


Fig. 2-5 Comparison between the human voice and the corresponding electronic voice.

Pseudorandom versus Periodic Noise

In linear predictions, an energy source is fed into a model of the vocal tract. In general, two energy sources are used—a *pseudorandom noise source* and a *periodic source* (see Fig. 2-5). The source which will be used to send sound frequencies into the filter is determined by whether the sound is voiced or unvoiced. For a voiced sound the voicing selector will feed in a periodic source (similar to the periodic vibration of the vocal cords). For an unvoiced sound (noiselike speech sounds) the selector will supply the filter with a pseudorandom noise source. This signal will then be multiplied by the amplitude factor (gain) to increase or decrease the volume of the signal. The signal then passes through the model of the human vocal tract and is filtered by it. This corresponds to the human vocal cavity filtering the signal generated by the vocal cords. Finally, the signal is passed through a digital-to-analog converter and fed into an audio amplifier and then a speaker. This signal is what is referred to as *synthetic speech.*

Reflection Coefficients

To understand the electronic filter network, we must first understand how the human vocal tract mechanism filters speech signals. The varying shape of the vocal tract can correspond to a series of pipes having different diameters. When waveforms pass from one pipe to the next, they generate waveforms in the opposite direction, known as *reflected waveforms.* These reflected waveforms have certain parameters associated with them known as *reflection coefficients.* The reflection coefficients, which correspond to the formants, or peaks, in the speech signal (see Fig. 2-6), are analyzed and used in various models of the vocal tract. Both the *lattice filter model* and the *cascade filter model* electronically simulate the characteristics of the human vocal tract. If we have one set of numbers that represents one position of the human vocal tract at any instant of time when the vocal tract changes shape, the set of numbers used in the filter model also changes. By varying this set of numbers to correspond to the vocal tract's changing shape, and feeding in the appropriate energy source to correspond

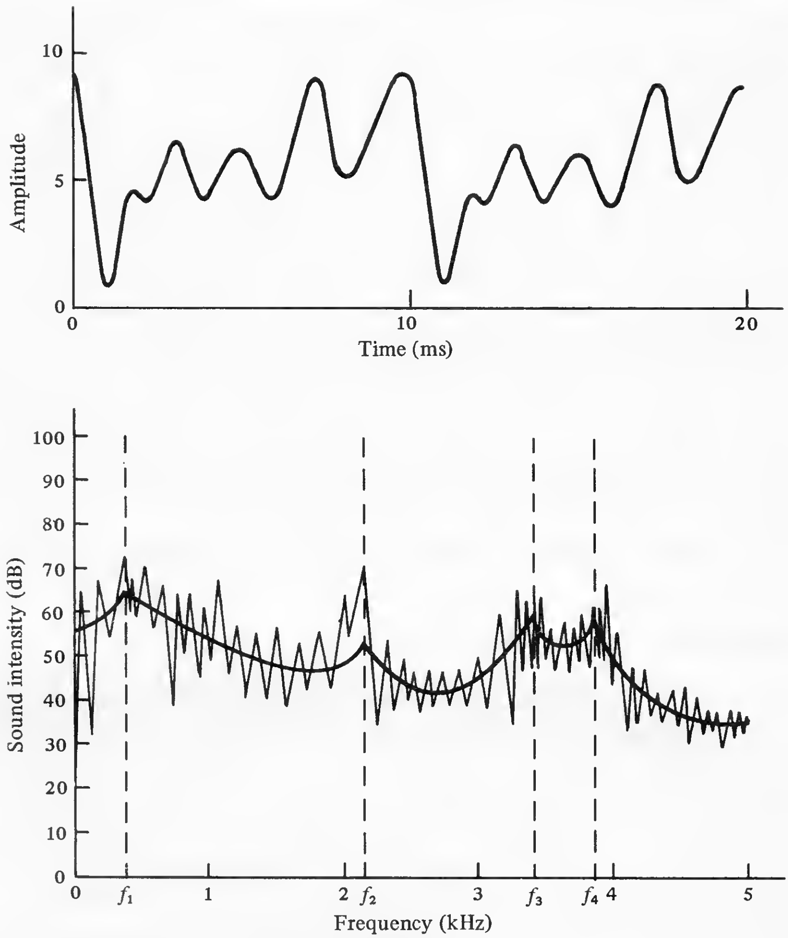


Fig. 2-6 Twenty milliseconds of voiced speech, the signal in the time domain (*top*) the signal in the frequency domain (*bottom*). The solid line is an LPC approximation of the signal. It is easy to see how the LPC approximation tends to reproduce the peaks or formants of the speech signal.

to a voiced or unvoiced portion of speech, synthetic speech can be created.

A typical sequence analysis of the LPC process contains four main operations.

1. After the speech signal has been passed through an analog-to-digital converter, it is broken into intervals, 15 milliseconds (ms) in duration, known as *analysis frames*. The short time energy of each frame is calculated and used to determine the amplitude or volume of the signal.
2. A voicing decision is made to determine whether the analysis frame is voiced or unvoiced. If the frame is classified as being voiced, the pitch period is determined.
3. An LPC analysis of each frame is performed. This produces a preselected number (usually 10 or 12) of reflection coefficients which best match the spectral characteristics of the sample analysis frame.
4. Next, the data for the appropriate vocal tract model used is coded. The generalized reflection coefficients are used to generate the actual filter coefficients that are applied to the vocal tract model. Adding the parameters of amplitude and pitch forms will result in a complete set of data known as a *synthesis frame*. This data is supplied to the filter and updated every 15–20 ms. If the frame of data includes a pitch parameter, the filter is excited with the periodic source. If there's no pitch parameter, the pseudorandom noise source is used.

The final speech data has now been reduced to approximately 2,000 bits per second, a level that offers an inexpensive means of storing synthetic speech data.

LET'S GET TECHNICAL . . .

LPC synthesizers incorporate all pole digital filters where the number of poles represents the number of formants in the speech signal. In this way at any instant in time the digital filter can have approximately the same frequency response as the human vocal tract. By substituting different poles or values for the digital filter representation, in relation to time, the model can now represent the varying shape of the human vocal tract (see Fig. 2-7).

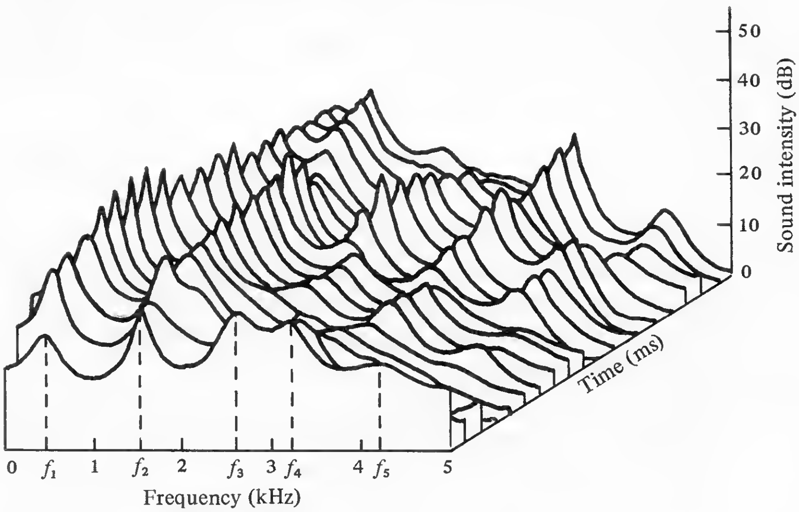


Fig. 2-7 A three-dimensional plot of the frequency response of an LPC synthesizer. At any instant of time the formants (f_1–f_5) can be analyzed. As the vocal tract changes shape the shape and location of the formants in the plot also change.

LATTICE FILTER MODEL

A lattice filter model uses the reflection coefficients, also known as the k parameters, of the speech signal. It can be described by the equation:

$$H(z) = \frac{g}{1 - \sum_{k=1}^{10} a_k z^{-k}}$$

For a 10-stage filter, a_k represents the 10 reflection coefficients, g is the gain, and z^{-k} represents the time delays. The lattice structure includes multiplication, summation, and delay blocks. This technique requires about 400,000 multiplications and additions each second (see Fig. 2-8).

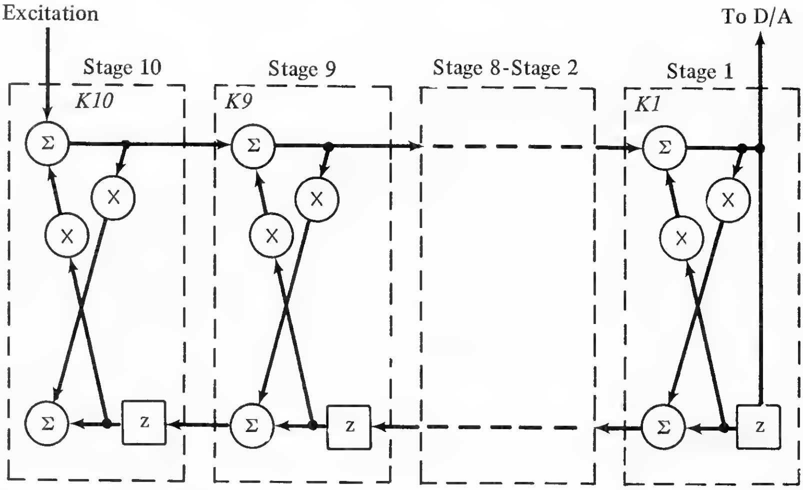


Fig. 2-8 A 10-stage lattice filter. This filter structure can simulate the resonant effects of the vocal tract.

CASCADE FILTER MODEL

A cascade filter model can consist of a series of second order sections. Each section is a digital resonator capable of modeling a single vocal tract resonance or formant. To model six formants, a 6-stage cascade filter is used forming a 12-pole digital filter (see Fig. 2-9 *top*). This system requires only one multiplication per pole whereas the lattice filter requires two multiplications per pole. This model uses frequency and bandwidth parameters which are derived from the reflection coefficients. Each pole of this system can be described by the equation:

$$H(z) = \frac{g}{1 - 2fz^{-1} - b_t z^{-2}}$$

where g represents the gain, f represents the formant center frequencies, and b represents the first formant bandwidths (see Figs. 2-9 *bottom* and 2-10).

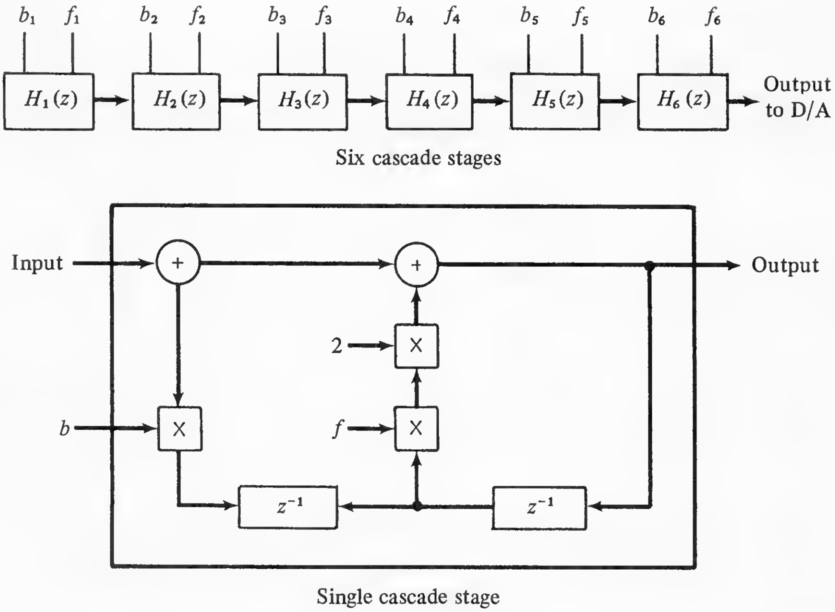
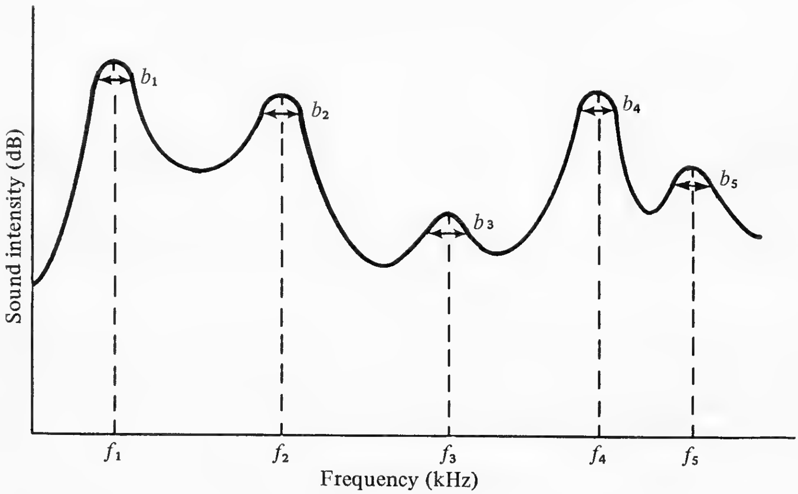


Fig. 2-9 By feeding the output of the first stage of a cascade filter into the input of the second stage, and so on for six stages, six formants of the speech signal can be simulated (*top*). Each stage can be used to represent one formant in the speech signal (*bottom*).

FORMANT CODING

The cascade filter is very advantageous in that it can also be used for *formant coding* techniques. Formant coding is similar to LPC in that it models speech signals in the frequency domain. It differs from LPC in that it uses only the center frequency values of the formants. The bandwidth values are either set to some constant value or are applied algorithmically. It further makes use of the fact that the intelligibility of human speech signals is found in the first three formants. For this reason, only the first three formant frequencies are used. Since this representation of the speech signal is less complete, less memory storage is required to store the speech data. Typically 600–800 bit per second rates can be obtained.



SPEECH PARAMETERS	DESCRIPTION
g	Gain
b_1	
f_1	
b_2	
f_2	
b_3	
f_3	Filter coefficients
b_4	
f_4	
b_5	
f_5	
b_6	
f_6	

Fig. 2-10 A speech signal in the frequency domain. Each formant has two parameters associated with it. One is a center frequency and the other is the bandwidth. These center frequencies and bandwidth values are specified by the filter coefficients used in a cascade model of the human vocal tract.

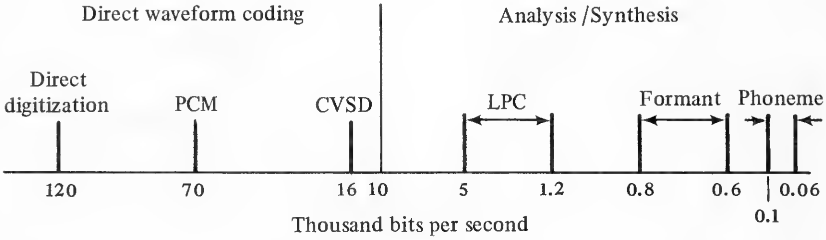


Fig. 2-11 Summary of the different techniques of speech synthesis and their related bit rates. Notice the drastic reduction in memory storage from the PCM technique, to the allophone technique. For this reason the cost of memory is reduced and you can now "Make Your Computer Talk" for under $35.

However, since the last three formants are not used (and it is this information that gives speech its emotion, quality, intonation, and emphasis), high-quality speech being recognizable of a particular speaker cannot be achieved.

SUMMARY

When choosing a synthesis reproduction technique, questions with regard to quality of speech and size of memory storage space must both be considered (see Fig. 2-11); however, these conditions may vary from application to application. For consumer-type applications where cost is critical, less memory is usually required and you would probably opt to use an LPC technique. In large industrial applications, where quality is the main constraint, the PCM technique may be more advantageous. Since there are so many ways to store synthesized speech, the choice is left to the system designer.

CHAPTER THREE

THE PHONEME/ ALLOPHONE APPROACH

GLOSSARY

ACOUSTIC PHONETICS The physical characteristics of a speech signal.

ALLOPHONE A variation of a particular phoneme sound that depends on its position in a word.

ARTICULATORY PHONETICS The process of making speech sounds or phonemes.

COARTICULATION The blending together of two allophones.

PHONEME The smallest unit of speech that is used to distinguish meanings between words.

SUBWORD UNITS

Although the PCM and LPC methods of encoding speech sounds discussed in Chapter Two give very high-quality speech, the memory requirements to store a number of words is very large. Use of these processes to store synthesized speech as whole units (i.e., words, sentences) makes the storage of a large vocabulary very expensive. Using these processes to store *subword units* (i.e., syl-

lables, half syllables, etc.), which can be combined to make whole words, large vocabularies can be stored in a small amount of memory. The words formed by combining these subword units will typically sound somewhat less understandable than words which are stored as whole units. However, if the set of subword units is designed properly, *any* word in a given language can be constructed from this set.

Phonemes

The smallest unit of speech is called a *phoneme*. Phonemes are actually smaller than syllables and even half syllables.

Consider the following example: The word "bee" is a monosyllabic (one syllable) word. However, the syllable "bee" is actually composed of two phonemes. The first phoneme is the sound of the letter **b** and has been named /b/. The second phoneme is the sound of the letters **ee** and is what we refer to as the long **e** sound. This sound is equivalent to the **ea** sound in "eat" and is named /e/.

Although a phoneme rarely appears as an entire word (e.g., /a/ as in **a** desk), a change in a single phoneme can change one word into another. For example:

WORD	*PHONEMES*
bee	/b/ /e/
pea	/p/ /e/

The phonetic symbols /p/ and /b/ are sufficiently different to signal a difference in meaning between the words "pea" and "bee"; however, the phonemes /p/ and /b/ mean nothing by themselves. Therefore, a phoneme does not typically have meaning but is used to distinguish meanings between words. This process of making speech sounds is known as *articulatory phonetics.*

Allophones

To compare with the above example, another word which uses the phonemes /b/ and /e/ is "bleed." In theory, the sounds /b/

and /e/ should sound exactly the same as the /b/ and /e/ sounds in "bee." However, the sounds we actually make when speaking are affected by the sounds that precede and follow the particular sound. In this case, the /b/ as well as the /e/ is affected by the sound of the letter l. It should also be noted that the /b/ and the /e/ affect the sound of the letter l. The phonemes can be thought of as "blending together" at their "edges." This blending does not create the addition of new phonemes. It actually creates a *variation* of a given phoneme, which is called an *allophone*. Let's look at our example:

WORD	*PHONEMES*
bee	/b/ /e/
bleed	/b/- /e/-

Since the sounds are slightly different from each other when pronouncing each word, the allophones may be represented as follows:

WORD	*ALLOPHONES*
bee	BB1 EE1
bleed	BB2 EE2

For another example, consider the word "Dad." The phonetic description of "Dad" would be: /d/ /ae/ /d/. The ideal /d/ sound would be considered a phoneme. However, the initial and final /d/ sounds are different; therefore, they can each be referred to as DD1 and DD2. DD1 and DD2 are allophones of the phoneme /d/. In essence, an allophone is the variation of a particular phoneme, depending upon its position in a word.

Coarticulation

We have just reviewed one characteristic of acoustic phonetics: the physics of the sound wave characteristics of speech signals. This was illustrated when comparing two words that use the same

phonemes. Another characteristic of a speech waveform arises when trying to extract discrete speech sounds from the continuously varying speech signal. During the pronunciation of a word, the articulators are constantly moving from one phoneme position to another. For this reason, each sound in the sequence influences every other sound around it. Because these speech sounds "overlap," as previously described, a problem arises when extracting a speech sound that is "pure" (i.e., a sound that does not contain cues to other speech sounds). For example, if you try to extract the **b** sound from the word "brain" by taking larger portions of the acoustic signal from the beginning of the word, one would encounter a nonspeechlike noise and then the sound **br**. There is no point at which the **b** sound can be heard in isolation. Due to this *coarticulation effect,* information is lost when extracting these sounds and the speech quality is reduced somewhat. Methods of improving this overlapping of sounds are discussed in the Let's Get Technical section at the end of this chapter.

When designing a set of subword units, as mentioned above, taking this coarticulation effect into account will result in more natural sounding speech. Each language has a set of subword units which is slightly different from that of other languages. There are approximately 42 phonemes in the English language. Each of the speech synthesizers described in Chapters Four through Eight use a set of 59 allophones derived from these 42 phonemes. From this set we can produce any word in the English language.

HOW TO CREATE WORDS FROM BASIC SOUNDS

We have just discussed a speech synthesis method known as *allophone synthesis* which can "make your computer talk." Utilizing these basic sounds of the English language, an unlimited vocabulary can be created for your personal computer. However, due to the coarticulation effects of allophone synthesis, the unlimited vocabulary has been obtained at the expense of speech that is not as natural or smooth as LPC or PCM methods. Allophone synthesis also requires familiarity with the *sounds* that form the words, which

are quite different from the *letters* that form the words. The symbols used to represent the allophones must also be studied. Table 3-1 gives a detailed set of guidelines for using the allophone set with the speech synthesizers described in the hardware/software sections of the book (see Chapters Four through Eight). For a further explanation of their classifications (i.e., vowels, resonants, etc.) see General Phoneme Classifications in the Let's Get Technical section of this chapter.

Table 3-1 ALLOPHONE GUIDELINES*

ALLOPHONE	SAMPLE WORDS†	DURATION (ms)
Short vowels‡		
IH	sitting, stranded	70
EH	extent, gentlemen, end	70
AE	extract, acting, hat	120
UH	cookie, full, book	100
AO	talking, song, ought	100
AX	lapel, instruct, succeed	70
AA	pottery, cotton, hot	10
Long vowels		
IY	treat, people, penny, see	250
EY	great, statement, tray, beige	280
AY	kite, sky, mighty	260
OY	noise, toy, voice, boy	420
UW1	after clusters with YY: computer	100
UW2	monosyllablic words: two, food	260
OW	zone, close, snow	240
AW	sound, mouse, down	370
EL	little, angle, gentlemen	190
R-colored vowels		
ER1	letter, furniture, interrupt	160
ER2	monosyllables: bird, fern, burn	300
OR	fortune, adorn, store	330
AR	farm, alarm, garment	290
YR	hear, earring, irresponsible	350
XR	hair, declare, stare	360
Resonants		
WW	we, warrant, linguist	180
RR1	initial position: read, write, x-ray	170
RR2	initial cluster: brown, crane, grease	120

Table 3-1 (*Continued*)

ALLOPHONE	SAMPLE WORDS†	DURATION (ms)
Resonants		
LL	like, hello, steel	110
YY1	clusters: cute, beauty computer	130
YY2	initial position: yes, yarn, yo-yo	180
Voiced fricatives		
VV	vest, prove, even	190
DH1	word-initial position: this, then, they	290
DH2	word-final and between vowels: bathe, bathing	120
ZZ	zoo, phase	210
ZH	beige, pleasure	190
Voiceless fricatives		
FF‡	food	150
TH‡	thin	180
SS‡	sit	90
SH	shirt, leash, nation	160
HH1	before front vowels: YR, IY, IH, EY, EH, XR, AE, he, hen, hit, hear, heat, hay, hair	130
HH2	before back vowels: UW, UH, OW, OY, AO, OR, AR, hue, hook, hoe, hoist, hawk	180
WH	white, whim, twenty	200
Voiced stops§		
BB1	final position: rib between vowels: fibber, bleed, brown	50
BB2	initial position before a vowel: beast	50
DD1	final position: played, end	70
DD2	initial position: down; clusters: drain	160
GG1	before high front vowels: YR, IY, IH, EY, EH, XR, guest	80
GG2	before high back vowels: UW, UH, OW, OY, AX, clusters: green, glue	30
GG3	before low vowels: AE, AW, AY, AR, AA, AO, OR, ER, medial clusters: anger; final position: peg	160

ALLOPHONE	SAMPLE WORDS†	DURATION (ms)
Voiceless stops§		
PP	**p**leasure, am**p**le, tri**p**	210
TT1	final clusters before SS: tes**t**s, i**t**s	100
TT2	all other positions: **t**est, s**t**reet	140
KK1	before front vowels: YR, IY, IH, EY, EH, XR, AY, AE, ER, AX initial clusters: **c**ute, **c**lown, s**c**ream	160
KK2	final position: spea**k** final clusters: tas**k**	190
KK3	before back vowels: UW, UH, OW, OY, OR, AR, AO initial clusters: **c**rane, **qu**ick, **c**lown, s**c**ream	120
Affricates§		
CH	**church**, fea**t**ure	190
JH	**judge**, in**j**ure	140
Nasal		
MM	**m**ilk, alar**m**, a**m**ple	180
NN1	before front and central vowels: YR, IY, IH, EY, EH, XR, AE, ER, AX, AW, AY, UW final clusters: ear**n**	140
NN2	before back vowels: UH, OW, OY, OR, AR, AA, **n**o	190
NG	stri**ng**, a**ng**er, a**n**chor	220
Silence		
PA1	before BB, DD, GG, JH	10
PA2	before BB, DD, GG, JH	30
PA3	before PP, TT, KK, CH, between words	50
PA4	between clauses and sentences	100
PA5	between clauses and sentences	200

* Taken from the General Instrument Application Report, 1982; reprinted with permission from General Instrument.

† Boldfaced letters indicate allophone sound.

‡ These allophones may be doubled for initial position and used singly in final position.

§ These allophones require a pause before saying the allophone.

ALLOPHONE GUIDELINES

The first column of Table 3-1 represents the allophone names (symbols) which will be typed into your computer to generate a particular sound. The second column gives sample words and shows how the allophone sounds are used in context. Here you can understand the actual sound each symbol represents. The third column shows the duration (in milliseconds) of each sound or allophone.

Initial Allophones versus Final Allophones

For some phonemes there are two allophones to account for the initial and final position. In final position, stop consonants are usually unreleased. For example, when pronouncing words such as ri**b**, playe**d**, and pe**g**, the final stop consonants (b,d,g) are shortened or not fully pronounced (unreleased) because they are not followed by other phonemes. For this reason, when using a stop consonant in the final position of a word, an allophone with a shorter duration is required. As a result, an allophone designed for an initial position may sound too loud or strong in the final position and vice versa. Notice that the initial version of some allophones is longer than the final version (see Table 3-1). For

POSITION	*ALLOPHONE*	*DURATION (ms)*
Initial	DD1	160
Final	DD2	70
Initial	DH1	290
Final	DH2	120

The allophones footnoted ‡ in Table 3-1 can be doubled or tripled. This means that these allophones may be joined together in succession. This feature is only incorporated in selected allophones. Therefore, to create an initial s you can use SS, SS as opposed to one SS at the end of a word. This can also be accomplished with the TH, FF, and the short vowels. Other phonemes

may appear as three different allopohnes. These allophones are also used for different vowel contexts.

Stressed versus Unstressed Allophones

Studies have shown that stressed syllables are higher in amplitude and pitch and longer in duration than unstressed syllables. Duration is the more prominent cue to stress, when compared with amplitude. Because of this fact, a syllable will sound stressed if its vowel is lengthened. For this reason, it is useful to double short vowels when stress of a particular sound is required. For example, in the word "is" (IH ZZ), you may want to double the IH allophone for increased stress, IH IH ZZ. You can also create differences between words such as the word "subject" (as a noun), which is stressed on the first syllable, and "subject" (as a verb), which is stressed on the second syllable. This can be accomplished by using AX twice in the first syllable of the noun and EH twice in the second syllable of the verb. For example:

WORD	*ALLOPHONE*
subject (noun)	SS SS **AX AX** PA2 BB1 PA2 JH EH PA3 KK2 PA3 TT2
subject (verb)	SS SS AX AX PA2 BB1 PA2 JH **EH EH** PA3 KK2 PA3 TT2

Long vowels cannot be doubled but the UW allophone appears with two durations. The short one, UW1, sounds good in words with many syllables after YY, as in com**pu**ter. The long version, UW2, is used for increased stress in monosyllabic words such as two and f**oo**d.

R-Colored Vowels

The column labeled "R-colored vowels" contains allophones created from the vowel sound plus **r.** The /er/ sound, in particular, contains two versions. The short one, ER1, is useful in words that end in **er** (let**ter**, bit**ter**). The long one, ER2, is used for increased stress in monosyllabic words (**fur**, **bird**).

Concatenation of Consonants and Vowels

Several phonemes have allophones that were specifically designed to *concatenate* or join together with other phonemes. For example, TT1 was designed to be used in final clusters before SS, as in its or tests. RR2 was designed to be used in initial clusters as in grease and breeze. YY1 is used in clusters with UW1. Because of the coarticulatory effects of vowels on some consonants, different allophone consonants are required depending on the vowel context. GG1 is needed before allophones such as IY, YR, IH, or EH (guest). NN2 is needed before allophones such as UH, OY, OR, or OW (no).

A Word on Pauses

Some sounds, labeled as voiced stops, voiceless stops, and affricates in Table 3-1, require a brief duration of silence before them. It has been shown that shortening the silent duration before a voiceless stop results in the perception of a voiced stop and the converse also holds true. Therefore, voiceless stops require a longer duration of silence or pause before them than voiced stops. So a PA1 or PA2 may be used before BB, DD, GG, and JH while a PA3 is used before PP, TT, KK, and CH. This will cause the following allophone to appear to be stressed somewhat. The allophones that require pauses before them appear in Table 3-1 (see table footnote §). You may need to change the duration of the pause a few times before it sounds right, but don't get discouraged, because it will soon become an automatic process to you!

Take a look at the following sample words. This will give you some insight into allophone synthesis.

WORD	*ALLOPHONE*
Hello	HH1 EH LL OW
I	AY
can	KK1 EH PA1 NN1
give	GG1 IH VV
the	TH IY

power	PP AW ER1
of	AX VV
speech	SS PP IY CH
to	TT2 UW2
all	AO AO PA1 LL
computers	KK1 AX MM PA3 PP YY1 UW1 PA3 TT2 ER1 ZZ
Your	YY2 OR
wish	WW IH IH SH
is	IH IH ZZ
my	MM AY
command	KK2 AX MM AE NN1 DD1

To create the word "computers," think of how it sounds, not the way it's spelled. Using Table 3-1, pick out the first sound, which is the KK1 allophone. KK1 was chosen because the next sound, when spoken slowly, sounds like the /a/ sound in l**a**pel. The allophone used to represent this sound is AX and KK1 is used before AX. The following sound is an /m/ sound so we will use the MM allophone. Now we have KK1 AX MM which represents the "com" in computers.

Next we must find a /p/ sound. Look under voiceless stops, and you will find a PP sound as in tri**p**. Because it is best to use a pause before voiceless stops, try adding a PA3 before PP. The next sound is a little tricky. One may think the sound is a /u/ sound. Well that's only half correct. If you look under resonants, you will find a YY1 sound and a YY2 sound. Since the YY1 sound is used in clusters as in c**u**te and b**eau**ty, this becomes the next allophone, because the word computer contains the cluster "pute." To continue the cluster we'll need the /u/ sound. You will find two choices under the long vowels. Notice UW1 is used after clusters with YY.

Now to continue the word, we need a /t/ and /er/ sound. First insert another PA3; remember a pause is used before a voiceless stop. Here, we have two possibilities, TT1 or TT2. Because the /t/ sound is not in a cluster with SS, the TT2 allophone will

be used. The next sound is a vowel sound followed by an /r/ sound. Remember there were special allophones designed for specific use in this case. They are called the R-colored vowels. Here again, we have two choices. "Computers" is surely not a monosyllabic word, so the only choice left is ER1. The final sound is also a little tricky. Once again, think of how the word *sounds* and not the way it's *spelled*. If you try an SS, you will find this is incorrect. The sound is really a /z/ sound. So let's end the word with the ZZ allophone. The final word should look like this:

KK1 AX MM PA3 PP YY1 UW1 PA3 TT2 ER1 ZZ

For practice, let's try to create the word "wish." You may not know whether to use the WW sound or the WH sound. The correct sound is the WW. To fully explain why this is true would be a lengthy explanation; however, the position of the lips, and how the air is expelled, are two major reasons why WW is the better choice. For this situation it would be easier to try both of them. Whichever one sounds better to you is the one to use. The next sound is the short /i/ sound as in sit. Among the short vowels there is an IH allophone. These short vowels are unique in that they can be stressed. This particular word requires a little stress, so double the IH allophone. (To hear the differences, try both ways.) The last sound is the /sh/ sound as in **sh**ip. Under voiceless fricatives, you will notice the SH allophone. The completed word is:

WW IH IH SH

For more examples of how to construct words from allophones, see Appendix A.

When constructing words with allophones, always remember to think about how a word sounds and not how it's spelled; in some cases this may be obvious, in other cases it may not. It's obvious that an NG allophone belongs at the end of the words so**ng** and lo**ng**. It's not so obvious that it is used to represent the /n/ sound in u**n**cle. Furthermore, some sounds may not even be represented in words as letters, like the YY in comp**u**ters.

Please note that the above suggestions are not rules. You may want to play with different sounds or different pauses to create a sound that is pleasing to your ear. (Don't be surprised if your

synthesizer has your regional accent!) Speech synthesis is so subjective that what one person likes, another may not. Because allophone synthesis gives the user the ability to change sounds at will, it provides a rich environment for experimentation. So, go ahead and change a few sounds. The intent of this book is to teach you the fundamental concepts of allophone synthesis.

LET'S GET TECHNICAL . . .

Before we take an in depth look at allophone synthesis, let's review a few levels of analysis in linguistics.

There are many levels of analysis in the study of how words are created and how they interact with each other when joined in sentences. We have already reviewed the lowest level, allophone synthesis. Let us now take a look at the other levels of analysis.

There are basically four levels of analysis when dealing with the structure, construction, and meaning of a word:

1. **Semantics** is the study of the development and changes in meanings of words. These changes occur depending on how the word is used in a phrase or sentence. For example, the word "dress" has two meanings. As a noun "the *dress* is blue" and as a verb "I will *dress* the baby."
2. **Syntax** is the relationship of arrangements of words used in phrases or sentences, with varying degrees of length and complexity. For example, if one says "the *man* bit the *dog*" it conveys a significantly different meaning from "the *dog* bit the *man*" even though the individual words used have exactly the same meaning in both cases.
3. **Morphology** deals with the construction of different types or classes of words. How nouns are made plural or how verbs are made into the past tense are two examples.
4. **Phonology** is the study of speech sounds. It includes areas concerning the distribution of sounds and how "neighboring" sounds affect one another.

IMPROVING COARTICULATION

Although each level of analysis has its own distinct meaning, in a given situation, they all interact with each other. The types of analysis just discussed make a text-to-speech system possible. (This system is discussed in detail in Chapter Ten). A text string (composed of normal English words) is entered and the computer system speaks it. For this type of system two sets of rules are required: The first set of rules converts the text entered into the computer to the appropriate allophone sounds. (An ideal system of this type would require a complex program incorporating rules for all four levels of analysis.) The second set of rules converts the allophone symbols to sounds used to pronounce the desired word or words. In this chapter we have just discussed the second set of rules using the phoneme/allophone approach.

The major advantage of the phoneme/allophone approach is that it can create an unlimited vocabulary for your computer from a limited inventory of sounds. However, as stated previously, the speech quality is reduced due to the overlapping of certain sounds (coarticulation effect), when compared to natural speech.

Diphones

One method of improving coarticulation is with the use of *diphones*. Diphones are sounds that encompass the transition from one sound to the next. They extend from the center of one phoneme to the center of the next phoneme.

Morphs

Another method to increase speech quality is with the use of *morphs*. Morphs are the smallest unit of sound that can convey meaning. They consist of root words, prefixes, and suffixes. For example:

WORD	*PREFIX*	*ROOT*	*SUFFIX*
unrelated	un	relate	ed
previewed	pre	view	ed

Demisyllables

Still another method is the *demisyllable* approach. Demisyllables consist of initial and final half syllables and phonetic affixes. For example:

WORD	*DEMISYLLABLES*	
box	bo	ox

A Good Compromise

Although diphones, morphs, and demisyllables increase the quality of allophone synthesis, sets of these tend to contain more units, so the memory required becomes prohibitive just as with LPC and PCM techniques. Even without the use of these extensions, allophone speech synthesis still offers a good compromise among many factors including versatility, flexibility, cost, hardware complexity, size of vocabulary, memory storage, and quality of speech.

GENERAL PHONEME CLASSIFICATIONS

The following list describes how phoneme sounds are classified. Their classification depends on how the sounds were produced, which articulators were used to produce the sounds, where the sounds were produced in the vocal cavity, and whether or not the vocal cords were vibrating.

Vowels:

Produced with a relatively unconstricted vocal tract. The energy source is the vibration of the vocal cords, which is periodic in nature. Vowels are classified according to whether the front or back of the tongue is high or low, whether they are long or short, and whether the lips are rounded or not rounded. They include the short, long, and R-colored vowels (see Table 3-1).

Consonants:

Produced by creating a constriction in the vocal tract where the source can be in place of or in conjunction with the vocal

cords. For stops, fricatives, and affricates, the energy source is aperiodic (pseudorandom). For others it may be periodic or a combination of periodic and aperiodic. Consonants are classified by the place and manner of articulation and by the articulatory features of voicing. The place of articulation refers to the point in the vocal tract where the sound is made or where two articulators make contact. The manner of articulation refers to how the consonants are made. It describes the way in which the articulators make contact to produce the speech sound. Voicing refers to whether the vocal cords are vibrating or not at the time the sound is produced. The resonants, fricatives, stops, affricates, and nasals are classified as consonants (see Table 3-1).

Voiced Phonemes:

Phonemes that are produced with the energy source being the vocal cords. (This includes all vowels, resonants, voiced fricatives, and voiced stops; see Table 3-1.)

Unvoiced Phonemes:

Produced when the vocal cords are not vibrating and the energy source is at the lips or teeth. (This includes voiceless fricatives, stops, and affricates; see Table 3-1.)

Resonants:

Formed by continuous movement of the tongue from an initial target to another point for the following vowel sound.

Fricatives:

Produced using a narrow constriction as the energy source and allowing a rush of air to flow through it.

Stops:

Produced by blocking the flow of air through the oral cavity, causing the air pressure to build up. When this pressure is released, a short burst of noise is generated.

Affricates:

Produced by first blocking the vocal tract entirely and then allowing air to flow through a narrow constriction. Affricates

can be considered as a combination of a stop consonant followed by a fricative.

Nasals:

Produced by blocking the oral cavity and allowing the air to pass through the nasal cavity.

SUMMARY

We've shown how unlimited vocabularies can now be added inexpensively to home computers. Words are stored as subword units, called phonemes and allophones, which occupy a relatively small amount of memory storage space. But speech quality suffers slightly, and so methods of improving coarticulation have been developed with the use of diphones, morphs, and demisyllables. However, the additional memory needed to store these extensions of allophones reduces its main advantage in the marketplace—its low cost. In conclusion, allophone synthesis offers the best compromise for most applications of speech to home computers.

CHAPTER FOUR

MAKING YOUR APPLE II, IIplus, AND IIe TALK

The Apple computer is a very useful computer with which to adapt synthesized speech. Because of its wide popularity, many software programs have been written to support it—now these programs can speak to you!

This chapter explains how you can make your Apple II, IIplus, and IIe talk. It consists of two sections: one describes the hardware interfaces, schematics, and parts required, while the other explains and provides the software required to drive the associated hardware. A detailed discussion of the program listing and its editing features are also included.

HARDWARE

This section explains all that's required to build a speech interface unit for your Apple computer. It is divided into three parts: the *Starter Kit,* the *Complete Kit,* and the *Complete Module.*

The starter kits for each particular model contain the components denoted in Table 4-1 by an asterisk. (They usually include the speech synthesis chip, the ceramic resonator, and the TTL components.) The remaining components (experimenter's breadboard, connector, ribbon cable, resistors, capacitors, etc.) have to be purchased separately. These parts are very popular and if you don't have them around your lab, any electronics store (e.g., Radio Shack) will surely have them. The starter kit is the least expensive type

Table 4-1 APPLE II, IIplus, AND IIe PARTS LIST

QUANTITY	PC BOARD IDENTIFICATION FOR THE COMPLETE KIT	DESCRIPTION
1	PC board not available at the time of this publication	PC board or protoboard
1		Reset switch—used as a hardware reset
1		Power jack (mini)
1		RCA phono speaker jack ($\frac{1}{4}''$)
1		10K potentiometer (volume control, 3-prong)
1		7805 C or LM340-5 5-V regulator
1		* SPO256-AL2 speech synthesizer
1		* 74LS00 TTL logic circuit
1		* 74LS02 TTL logic circuit
1		* 74LS368 TTL logic circuit
1		LM386–1 operational amplifier (This is required to drive a 4- or 8-Ω speaker.)
1		28-pin socket (for the SPO256-AL2)
2		14-pin sockets (for the 74LS00 and 74LS02)
1		16-pin socket (for the 74LS368)
1		8-pin socket (for the LM386)
1		DPDT switch (on/off internal power)
1		* 3.12-MHz ceramic resonator (blue, CSA3.12MS2)

QUANTITY	PC BOARD IDENTIFICATION FOR THE COMPLETE KIT	DESCRIPTION
4 Resistors:		
2		33-kΩ (orange, orange, orange)
1		10-kΩ (brown, black, orange) or 100-kΩ (brown, black, yellow)
1		10-Ω (brown, black, black) or 11-Ω (brown, brown, black)
12 Capacitors:		
3		0.1 μF
1		100-μF electrolytic (audio filter)
1		10-μF electrolytic (power filter)
1		1-μF electrolytic
1		10-μF electrolytic (audio filter)
1		100-μF electrolytic (power filter)
2		0.025 μF
2		100 pF
Additional parts required (*not included*)		
1		7.5 to 9-V dc power supply (250–300mA)
		Optional—to be used only if external power is required. Can be obtained at Radio Shack, Catalog #2731455 or Catalog #603053.
1		Speaker—any 4- or 8-Ω speaker may be used; the choice is yours.

* The only parts included in the starter kits.

of kit; however, it requires the most amount of time to build the circuit. If you are not familiar with wiring up breadboards, you may want to try the complete kit or complete module. The starter kits were designed for the more experienced hobbyist who wants to save a little extra money in exchange for a portion of his or her time.

The complete kits consist of all the parts listed in Table 4-1, including printed circuit (PC) boards, connectors, and cabinet. The speaker (and possibly the power supply) is the only part that has to be supplied. This kit is somewhat more expensive, but offers the ease of building the circuit on a PC board specifically designed for your particular computer. No wiring experience is necessary. All you need to know is how to use a soldering iron. This circuit is designed for the beginner who wants to learn how to build his or her own circuit and at the same time save a little money.

If after reading the instructions you are not interested in building a synthesizer yourself, you can purchase a complete module specifically designed for your computer. The complete modules are fully assembled and tested, and plug directly into the various computers, enabling them to speak in a matter of minutes!

> NOTE: The parts that form the starter kits, complete kits, and complete modules can be purchased from a number of distributors listed in the Parts Supplier Listing, Appendix B. With some distributors you may have to purchase all the parts separately, with others you may be able to purchase the parts in kit or module form. (Refer to this listing and note the parts that each company supplies before setting out to purchase your speech synthesis kits.)

All circuits, kit or module form, conform to certain requirements; they are discussed below.

PORT LOCATIONS AND POWER SPECIFICATIONS

This board may be plugged into slots 1 through 7. (Do not use slot 0.)

Apple IIplus and IIe

The base address for slot 1 is B = 49408 (decimal); for slot 2 it is B = 49664. For each additional slot, add 256 to the previous number. For example:

SLOT	*BASE ADDRESS (DECIMAL)*
3	49920
4	50176
5	50432
6	50688
7	50944

Apple II

The base address for slot 1 is B = −16128 (decimal); for slot 2 the base address would be −16128 + 256, or −15872. Again, 256 must be added for each additional slot. For example:

SLOT	*BASE ADDRESS (DECIMAL)*
3	−15616
4	−15360
5	−15104
6	−14848
7	−14592

The total amount of power supplied to the Apple bus is 500 mA. The speech synthesizer requires 250 mA. So, if any other peripheral is used which draws more than 250 mA, the speech synthesizer must be powered externally and the internal power must be disconnected.

PARTS LISTING

The list in Table 4-1 describes all parts required to build a speech synthesizer for your computer. Recall that the parts denoted by

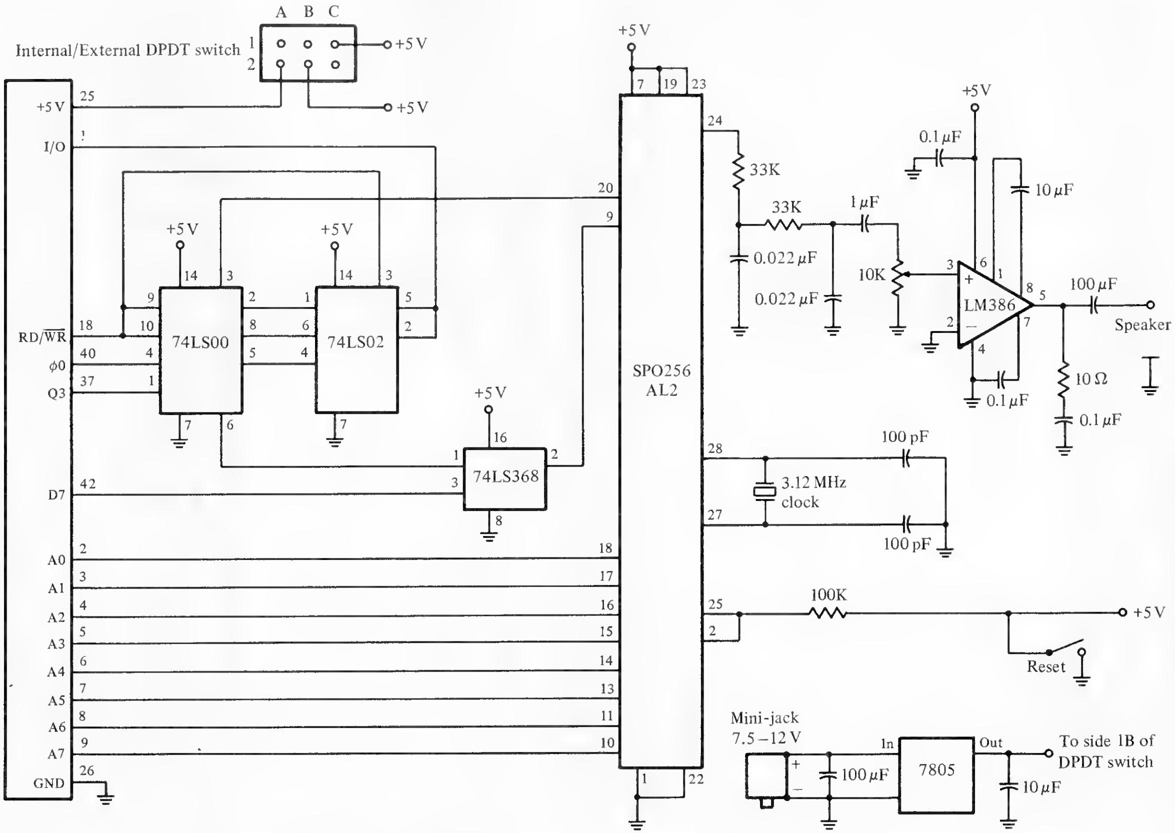


Fig. 4-1 Apple II, IIplus, and IIe speech circuit schematic.

an asterisk combine to form the starter kits. All additional parts must be supplied and wired up on an experimenter's breadboard. The complete kits include all the parts in the list with the exception of a speaker and power supply. The PC board identification (included for complete kits only) describes where the parts should be placed on the PC board (see the section on Complete Kit Assembly Instructions).

STARTER KIT ASSEMBLY INSTRUCTIONS

When building the starter kit, refer to the schematic diagram in Fig. 4-1.

NOTE: After completing your starter kit, see the section on Complete Module and Operating Instructions.

STEP 1 In addition to the starter kit, obtain all of the components listed in the Apple II, IIplus, and IIe Parts List, Table 4-1.

NOTE: A prototype breadboard, specifically designed for the Apple, that plugs directly into the bus is suggested.

STEP 2 Insert the sockets into the breadboard and make a note of where pin 1 is to be located. (Pin 1 should be in the upper left-hand corner.)

STEP 3 Solder or wirewrap the following connections:

FROM THE APPLE CARD:	*TO:*
+5 V pin 25	Side 2A of the DPDT switch
GND pin 26	Common ground of the synthesizer board
A0 pin 2	A1 pin 18 of the SPO256-AL2
A1 pin 3	A2 pin 17 of the SPO256-AL2
A2 pin 4	A3 pin 16 of the SPO256-AL2
A3 pin 5	A4 pin 15 of the SPO256-AL2
A4 pin 6	A5 pin 14 of the SPO256-AL2

FROM THE APPLE CARD:	*TO:*
A5 pin 7	A6 pin 13 of the SPO256-AL2
A6 pin 8	A7 pin 11 of the SPO256-AL2
A7 pin 9	A8 pin 10 of the SPO256-AL2
RD/$\overline{\text{WR}}$ pin 18	Pins 9 and 10 of the 74LS00 and pin 3 of the 74LS02
ϕ_0 pin 40	Pin 4 of the 74LS00
Q3 pin 37	Pin 1 of the 74LS00
I/O (Input/Output) pin 1	Pins 2 and 5 of the 74LS02
D7 pin 42	Pin 3 of the 74LS368

STEP 4 Next solder or wirewrap these connections:

FROM THE 74LS00:	*TO:*
Pin 6	Pin 1 of the 74LS368
Pin 2	Pin 1 of the 74LS02
Pin 3	Pin 20 of the SPO256-AL2
Pin 5	Pin 4 of the 74LS02
Pin 8	Pin 6 of the 74LS02

FROM THE 74LS368:	*TO:*
Pin 2	Pin 9 of the SPO256-AL2

STEP 5 The Power and Ground Connections

FROM SIDE 2B AND 1C OF THE DPDT SWITCH	*TO:*
	Pins 7, 19, and 23 of the SPO256-AL2
	Pin 14 of the 74LS00
	Pin 14 of the 74LS02
	Pin 16 of the 74LS368

FROM PIN 26 OF THE APPLE CARD	*TO:*
	Pins 1 and 22 of the SPO256-AL2
	Pin 7 of the 74LS00
	Pin 7 of the 74LS02
	Pin 8 of the 74LS368

STEP 6 The Reset Circuit

a. Connect a 100-kΩ resistor between +5 V (side 2 of the SPST switch) and pins 2 and 25 of the SPO256-AL2.

b. Connect the reset switch between ground (pin 26 of Apple card) and pins 2 and 25 of the SPO256-AL2.

STEP 7 The Regulator Circuit

a. Connect the positive side of the power jack to the 9-V input side of the 7805 (or LM340).

b. Connect the negative side of the power jack to the negative terminal of the 7805 (or LM340).

c. Connect the negative terminal of the 7805 (or LM340) to pin 26 of the Apple card.

d. Connect the 5-V output of the 7805 (or LM340) to side 1B of the DPDT switch.

e. Connect the positive side of a 100-μF electrolytic capacitor between the 9-V input of the 7805 (or LM340) and the negative side to ground (pin 26 of the Apple card).

f. Connect the positive side of a 10-μF electrolytic capacitor between the 5-V output of the 7805 (or LM340) and the negative side to ground (pin 26 of the Apple card).

STEP 8 The Oscillator Circuit

a. Connect the 3.12-MHz ceramic resonator between pins 27 and 28 of the SPO256-AL2.

b. Connect a 100-pF capacitor between pin 27 of the SPO256-AL2 and ground (pin 26 of the Apple card).

c. Connect a 100-pF capacitor between pin 28 of the SPO256-AL2 and ground (pin 26 of the Apple card).

STEP 9 The Audio Filter Circuit

a. Connect side 1 of a 33-kΩ resistor to pin 24 of the SPO256-AL2.
b. Connect side 2 of the first 33-kΩ resistor to side 1 of the second 33-kΩ resistor and to side 1 of the first 0.022-μF capacitor. Connect side 2 of the first 0.022-μF capacitor to ground.
c. Connect side 2 of the second 33-kΩ resistor to the positive side of the 1-μF electrolytic capacitor and to side 1 of the second 0.022-μF capacitor. Connect side 2 of the 0.022-μF capacitor to ground.
d. Connect the negative side of the 1-μF electrolytic capacitor to side 1 of the 10K potentiometer.
e. Connect side 2 (middle terminal) of the 10K potentiometer to pin 3 of the LM386.
f. Connect side 3 of the 10K potentiometer to ground.
g. Connect pins 2 and 4 of the LM386 to ground.
h. Connect a 0.1-μF capacitor between pins 4 and 7 of the LM386.
i. Connect the positive side of the 10-μF electrolytic capacitor to pin 1 of the LM386 and the negative side to pin 8 of the LM386.
j. Connect pin 6 of the LM386 to +5 V.
k. Connect a 0.1-μF capacitor between pin 6 of the LM386 and ground.
l. Connect pin 5 of the LM386 to side 1 of a 10-Ω resistor. Connect side 2 of the 10-Ω resistor to side 1 of a 0.1-μF capacitor. Connect side 2 of the 0.1-μF capacitor to ground.
m. Connect the positive side of the 100-μF electrolytic capacitor to pin 5 of the LM386. Connect the negative side of the 100-μF capacitor to side 1 of the speaker jack. Connect side 2 of the speaker jack to ground.

STEP 10 Insertion of the Integrated Circuits

NOTE: When inserting the IC's, be sure that pin 1 of the IC lines up with pin 1 of the socket that was labeled in step 2.

a. Insert the SPO256-AL2 into the 28-pin socket.
b. Insert the 74LS00 into the 14-pin socket (the socket connected to RD/$\overline{\text{WR}}$, ϕ_0, and Q3 of the Apple card).
c. Insert the 74LS02 into the 14-pin socket (the socket connected to I/O (input/output) of the Apple card).
d. Insert the 74LS368 into the 16-pin socket.
e. Insert the LM386 into the 8-pin socket.

STEP 11 Skip to the section on the Complete Module and Operating Instructions.

COMPLETE KIT ASSEMBLY INSTRUCTIONS

At the time this manual was printed, the complete kit assembly instructions were being developed and are unable to be included. For more information see the Parts Supplier Listings, Appendix B.

COMPLETE MODULE AND OPERATING INSTRUCTIONS

Whether you have built the starter kit or complete kit, or purchased the complete module, you are now ready to operate your speech synthesizer (see Fig. 4-2).

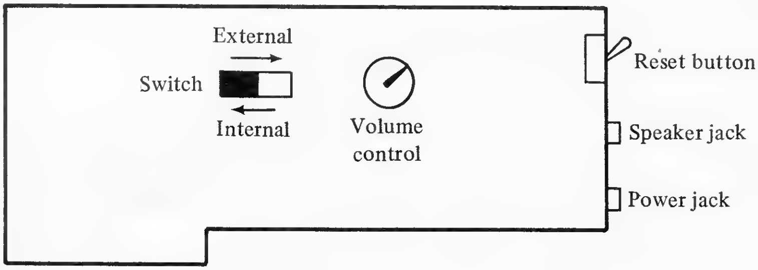


Fig. 4-2 Apple II, IIplus, and IIe component placement.

Reset Button—Depressing this button readies the synthesizer for operation. It will also cause the board to stop talking.

Volume Control—Turning this control to the right or left will increase or decrease the volume of the synthesizer.

Speaker Jack—This connection will drive any 4- or 8-Ω speaker or act as an auxiliary input to any receiver.

Power Jack—Optional power connection to be used for expanded system operation only (see the section on Port Locations and Power Specifications at the beginning of this chapter). This allows you to supply the additional power required if the other modules are being used. The power supply required is a 9-V, 300-mA supply, mini-jack (the tip is positive).

CAUTION: When using external power the internal/external switch should be in the external position. This disables the internal power circuit.

Internal/External Switch—While in the internal position, the synthesizer board is powered internally from the Apple computer. This is disconnected in the external position.

Operating Procedure

Using the Synthesizer with Internal Power (External Power Supply Not Required)

STEP 1 Switch the internal/external switch to the external position. This disables the internal power circuit.

STEP 2 Plug the speaker into the speaker jack.

CAUTION: The computer must be off.

STEP 3 Plug the synthesizer into any slot except slot 0. Remember the base address for slot 1 for the Apple IIplus and IIe is B = 49408, and for the Apple II it is B = −16128 (see the Port Locations and Power Specifications section at the beginning of this chapter).

STEP 4 Turn your computer on.

STEP 5 Turn the synthesizer switch to the internal power position.
STEP 6 Depress the reset button.

NOTE: A faint "click" should be heard in the speaker when depressing this button. If this sound is not heard, turn up the volume and try it again.

STEP 7 Skip to the section on Testing Your Circuit.

Using the Synthesizer with External Power

STEP 1 Switch the internal/external switch to the external position. This disables the internal power circuit.
STEP 2 Plug the speaker into the speaker jack.
STEP 3 Plug the external power supply into the power jack of the synthesizer board (the tip is positive).

NOTE: Do not plug into the wall outlet yet.

CAUTION: The computer must be off.

STEP 4 Plug the synthesizer into any slot except slot 0. Remember the base address for slot 1 for the Apple IIplus or IIe is B = 49408, and for the Apple II it is B = −16128 (see the Port Locations and Power Specifications section at the beginning of this chapter).
STEP 5 Plug in the 9-V power adapter to the synthesizer into the wall outlet.
STEP 6 Turn your computer on.
STEP 7 Depress the reset button.

NOTE: A faint "click" should be heard in the speaker when depressing this button. If this sound is not heard, turn up the volume and try it again. If condition still exists, proceed to the following section.

TESTING YOUR CIRCUIT

After carefully wiring up your speech synthesizer board, the following commands can be used to test your circuit before loading in the respective programs. Before power is applied, visually inspect

your hardware to ensure that the proper connections have been made and all the grounds are secure. On power up, a hardware reset is required—simply close the switch momentarily. A click or pop should be heard in the speaker. If this occurs, proceed with the instructions in this section. If this condition does not occur, refer to the section on Debugging Your Hardware at the end of this chapter.

Apple II

If you are using slot 1, the base address is B = −16128 (decimal); for each additional slot, add +256 to −16128. A simple POKE −16123,0 will speak the first allophone at address 5 (see Table 9-1). This is the OY allophone. If everything is correct, the synthesizer will continue to speak this allophone until a pause is entered. To enter a pause and silence the board, simply enter the statement POKE −16128,0. This is a pause at location 000. You are now ready to load the Exclusive Program in the usual manner (as specified in your computer manual) and create your own phrases.

Apple IIplus and IIe

If you are using slot 1, the base address is B = 49408 (decimal); for each additional slot add +256 to 49408. A simple POKE 49413,0 will speak the first allophone at address 5 (see Table 9-1). This is the OY allophone. If everything is correct, the synthesizer will continue to speak this allophone until a pause is entered. To enter a pause and silence the board, simply enter the statement POKE 49408,0. This is a pause at location 000. You are now ready to load the Exclusive Program in the usual manner (as specified in your computer manual) and create your own allophone phrases.

SOFTWARE

The Exclusive Phrase Finder Programs that follow allow you to build words and phrases from their constituent allophones. The

phrase can be edited by moving a pointer left or right to the desired position (see Table 4-2). Inserting, deleting, or replacing allophones can then be accomplished quite easily. When the phrase is prepared to your satisfaction, a simple ENTER, NEW LINE, or RETURN will signal the synthesizer to talk.

Table 4-2 ALLOPHONE EDITING COMMANDS

AVAILABLE COMMANDS	OPERATION
"phoneme strings"	Causes named allophone to be added to the phrase at the current position of the pointer, by either replacing the existing allophone or inserting one before it (see the section on Inserting an Allophone).
"L"	Moves the position pointer left one allophone.
"R"	Moves the position pointer right one allophone.
"D"	Deletes allophone at the current position pointer.
NEW LINE, ENTER, or RETURN	Causes the system to output to the hardware the commands necessary to pronounce the phrase.
"E"	Exits the program.
For the Apple IIplus and IIe only:	
"XL"	Moves the position pointer "X" number of allophones to the left.
"XR"	Moves the position pointer "X" number of allophones to the right.
"I"	Turns on the "insert mode." The next allophone entered will be inserted into the phrase at the current position of the pointer. Additional allophones will be inserted until "I" is entered again. The second "I" command will turn the insert mode off. When the insert mode is off, an entered allophone will replace the one at the current position. This is the default at system start-up.

Table 4-2 (*Continued*)

AVAILABLE COMMANDS	OPERATION
For the Apple II only:	
"XL and XR"	Commands are not applicable.
"I"	Turns on the "insert mode." The next allophone entered witll be inserted into the phrase at the current position of the pointer. Then the insert mode will automatically be turned off. When the insert mode is off, an entered allophone will replace the one at the current position. This is the default at system start-up.

EXCLUSIVE PHRASE FINDER PROGRAM DESCRIPTION

At system start-up, a brief message will be spoken and the following commands are performed.

> NOTE: The spoken message is deleted for the Apple II. For the Apple IIplus and IIe, this message will be spoken each time the program is RUN. To delete this message from these systems, the command "Delete line 75 from the program" must be issued.

The screen is cleared and all variables are initialized. The allophone symbol array is initialized with the 64 two- or three-character symbols that represent each allophone. These are strings the user will enter in order to add an allophone to the phrase (see Tables 3-1 and 9-1). The position pointer is at position one, and the user is prompted for input with a "≥" for the Apple IIplus and IIe. For the Apple II, the user is prompted with a "-." At this prompt, an allophone or any of the commands in Table 4-2 may be entered. An invalid allophone will be flagged as an error, as will attempting to move the position pointer to the left or right of the boundaries of the phrase. After each command, the updated phrase is displayed with the current position indicated by "≥" ("-" for the Apple II).

For example, at system start-up, the screen will look like this:

COMMAND	*SCREEN*	*COMMENTS*
RUN	⋗ ?	Pointer at position one.
HH1	⋗ ?HH1	Desired allophone.
ENTER	HH1⋗ ?	The first allophone has been entered; the pointer is at position two; the system is ready for the next input.
EH1	HH1⋗ ?EH1	Next allophone.
ENTER	HH1⋗ ?EH1 ***INVALID ENTRY***	User entered invalid data.
EH	HH1⋗ ?EH	
ENTER	HH1 EH⋗	The second allophone has been entered; the pointer is at position three; the system is ready for the next input.

Editing Features

Upon entering a string of allophones and noticing that a few corrections are in order, the following edit commands are useful.

First, we must position the pointer at the location where an editing command is to be performed. Let's take the word "hello" for example. The screen should now look like this:

HH1 EH LL UW1≥

Realizing that the UW1 allophone is incorrect, we would like to REPLACE it with "OW." To do so, the following commands are required:

Replacing an Allophone

COMMAND	*SCREEN*	*COMMENTS*
"L"	HH1 EH LL UW1≥ L	We have to move the pointer left one space to replace UW1.
"ENTER"	HH1 EH LL≥ UW1	The system is now ready to replace the allophone UW1.
"OW"	HH1 EH LL≥ UW1 OW	The desired replacement allophone is typed.
"ENTER"	HH1 EH LL OW≥	After pressing ENTER, the allophone has been replaced.

Moving the position pointer right works in the same manner as moving to the left. The only exception is that we use an "R" instead of an "L." These commands move the pointer one space at the time.

In replace mode (the default at program start-up), the new allophone will replace the allophone at the current position in the phrase.

NOTE: If you attempt to move the position pointer left or right beyond its boundaries, the message ***INVALID ENTRY*** will appear.

COMMAND	*SCREEN*	*COMMENTS*
"R" ENTER	HH1 EH LL OW> R ***INVALID ENTRY***	To clear the invalid entry, press ENTER or type in a valid command.
"L" ENTER	>HH1 EH LL OW L ***INVALID ENTRY***	

The XL and XR commands for the Apple IIplus and IIe work in the same manner as the L and R commands. The only difference is that these commands move the position pointer "X" number of allophones (or spaces) to the left or right. Once again, exceeding the boundaries will prompt an ***INVALID ENTRY*** message. For example:

COMMAND	*SCREEN*	*COMMENTS*
"2L"	HH1 EH LL OW1>	
ENTER	HH1 EH> LL OW	The position pointer has moved 2 spaces to the left.

Deleting an Allophone. We'll use the same examples as above. Remember, we must first position the pointer to the specified allophone to be deleted. Once this is accomplished, the following commands are required.

COMMAND	*SCREEN*	*COMMENTS*
"L" ENTER (3 times)	HH1 EH EH LL OW>	We need to delete an EH here, so first we must move the pointer 3 spaces to the left.
"D"	HH1 EH> EH LL OW D	The pointer is now positioned to the allophone to be deleted.

COMMAND	*SCREEN*	*COMMENTS*
"ENTER"	HH1 EH≥ LL OW	The EH allophone has been deleted.

NOTE: The delete command "D" deletes one allophone at a time.

Inserting an Allophone for the Apple IIplus and IIe. After creating your allophone phrases, and realizing that a few pauses (or other allophones) need to be inserted, the following command sequence must be performed. Let's use the word "chatter" as an example.

COMMAND	*SCREEN*	*COMMENTS*
"L" ENTER (3 times)	CH AE TT2 ER1 PA3≥ ?	We must position the pointer at the location where the inserted allophone will go.
"I"	CH AE≥ TT2 ER1 PA3 ?I	We are ready to turn on the "insert mode."
"ENTER"	CH AE≥ TT2 ER1 PA3 ?	Insert mode has been turned on.
"PA3"	CH AE≥ TT2 ER1 PA3 ? PA3	The desired allophone to be inserted is typed.
"ENTER"	CH AE PA3≥ TT2 ER2 PA3 ?	The allophone has been inserted. Additional allophones may be entered at this time if required.
"I"	CH AE PA3≥ TT2 ER2 PA3 ?I	We are ready to turn off the insert mode.

COMMAND	*SCREEN*	*COMMENTS*
"ENTER"	CH AE PA3> TT2 ER2 PA3 ?	Insert mode has been turned off.

Inserting an Allophone for the Apple II. When inserting an allophone, position the pointer at the location where the inserted allophone will go. Inserting an allophone can be regarded as combinations of insert and replace commands. The following commands must then be performed:

COMMAND	*SCREEN*	*COMMENTS*
"I"	HH1 LL OW	We will insert an allopone before LL.
"ENTER"	HH1>LL LL OW	The allophone LL has been repeated at the location of the desired insertion.
"EH"	HH1>LL LL OW	Desired allophone to be inserted.
"ENTER"	HH1 EH>LL OW	The EH allophone has effectively been inserted before LL and insert mode is automatically turned off. We are now in the replace mode again. To insert another allophone, we must turn on insert mode again with the same command sequence.

MAKING YOUR COMPUTER TALK

A simple "ENTER," "RETURN," or "NEW LINE" is all that's required to make the system talk. If a new allophone string is desired, you must first EXIT (E) the program and then RUN it again. This will clear all the previous allophone codes stored. Unfortunately at the time of writing this book, a "SAVE" routine was not supplied with the Exclusive Phrase Finder Program. You may want to try writing your own subroutine to save the allophone strings you've created with the program (in separate files). An alternative method is to save the Exclusive Phrase Finder Program on cassette tape or disk, with the phrase or word you've just created. When reloading the program, the following commands are necessary for it to speak the prestored phrase.

Apple II	GOTO 700
Apple IIplus and IIe	GOTO 300

Upon typing this command, the computer will speak the prestored allophone string and the allophone codes will be printed on the screen. The system is now ready to edit the existing string. To enter a new string, RUN the program.

EXCLUSIVE PHRASE FINDER PROGRAM LISTINGS

Apple II

NOTE: When typing in the allophones, it is essential to type blank spaces in place of all **b**'s in the program below. (The computer will only read allophones in groups of three letters or spaces.)

```
 0      CALL -936
 10     FOR X = 1 TO 1500: NEXT X: CALL -936
 12     B = -16128:  REM  Base Address for Slot 1
 15     DIM NA$(255):  DIM NB$(255):  DIM NC$(255)
 17     NA$ = "PA1PA2PA3PA4PA5OYbAYbEHbKK3PPbJHbNN1IHb
        TT2RR1AXbMMbTT1DH1"
 18     NB$ = "IYbEYbDD1UW1AObAAbYY2AEbHH1BB1THbUHb
        UW2AWbDD2GG3VVbGG1SHbZHbRR2FFbKK2KK1ZZb"
 19     NC$ = "NGbLLbWWbXRbWHbYY1CHbER1ER2OWbDA2SSb
        NN2HH2ORbARbYRbGG2ELbBB2"
 20     NA$(LEN(NA$)+1) = NB$
 21     NA$(LEN(NA$)+1) = NC$
 24     DIM I$(10):P = 1
 25     DIM PH(63)
 30     DIM S$(255)
 35     DIM T$(255)
 37     CU = 0
 40     GOTO 105
100     GOSUB 900
105     VTAB 7:PRINT
107     INPUT I$
108     VTAB 20: PRINT "Note: allow 20 spaces here"
110     IF I$ = "L" THEN GOTO 300
112     IF I$ = "R" THEN GOTO 400
114     IF I$ = "I" THEN GOTO 500
116     IF I$ = "D" THEN GOTO 600
118     IF I$ = "" THEN GOTO 700
120     IF I$ = "E" THEN GOTO 1000
140     GOTO 800
300     REM * Move Cursor Left *
305     IF P<2 THEN GOTO 950
310     P=P-1
330     GOTO 100
400     REM * Move Cursor Right *
405     IF P>62 THEN GOTO 950
407     IF P>L THEN GOTO 950
410     CU = 1
415     GOSUB 200
420     GOTO 100
500     REM *INSERT*
502     CU = 0
503     IF P>L THEN GOTO 950
505     IF L>62 THEN GOTO 950
507     PH(0) = PH(1)
510     FOR X = (L+1) TO (P+1) STEP -1
520     PH(X) = PH(X-1)
525     NEXT X
530     L = L+1
585     GOSUB 900
590     GOSUB 200
595     GOTO 100
600     REM *DELETE*
```

```
610     IF P>L THEN GOTO 950
615     IF L>62 THEN GOTO 950
620     FOR X = P TO L-1
630     PH(X) = PH(X+1)
635     NEXT X
640     L = L-1
650     CU = 0
660     GOSUB 900
670     GOSUB 200
680     GOTO 100
700     REM *SPEAK*
710     FOR X = 1 TO L
720     Z = B+PH(X)-1
730     IF PEEK (B)<128 THEN GOTO 730
740     POKE Z,0
750     NEXT X
760     IF PEEK(B)<128 THEN GOTO 760
770     POKE B,0
790     GOTO 100
800     REM *NAME DECODE*
805     IF LEN(I$) = 2 THEN I$(3) = "b"
807     Z = 0
810     FOR X = 1 TO 190 STEP 3
812     Y = X+2
813     Z = Z+1
814     IF I$ = NA$(X,Y) THEN GOTO 825
816     NEXT X
818     GOTO 950
825     PH(P) = Z
827     IF P>L THEN L = L+1
880     P=P+1
895     GOTO 100
900     REM*PHONEME LIST DISPLAY*
902     CALL -936
903     VTAB 12
904     IF P=1 THEN PRINT "-> ";
905     FOR X = 1 to L
910     Z = PH(X)*3-2:Y=Z+2
920     PRINT NA$(Z,Y) ; "b";
925     IF X+1=P THEN PRINT "-> ";
927     NEXT X
945     RETURN
950     VTAB 20: PRINT "INVALID ENTRY"
952     GOTO 105
1000    END
2000    PRINT P;"b";L;"b"
2010    Y = X+2 : Z=PH(X)
2020    PRINT NA$(Z,X)
2030    NEXT X
2040    END
```

Apple IIplus and IIe

NOTE: When typing in the allophones, it is essential to type a blank space in place of all **b**'s in the program below.

```
  0     HOME
  2     PRINT "***PHONEME CONCATENATION SYSTEM***"
  5     HOME
 10     DIM NA$(64) : REM PHONEME SYMBOL ARRAY
 12     DIM PH$(64) : REM CURRENT PHONEME STRING
 14     DIM PH(64)  : REM CURRENT PHONEME STRING ADDRESSES
 16     S$ = ""     : REM DISPLAY STRING, WITH CURSOR POSITION
 18     I$ = ""     : REM INPUT STRING FROM USER
 25     P=1         : REM CURRENT POSITION POINTER
 30     L=0         : REM LENGTH OF PHONEME STRING
 35     IN=0        : REM "INSERT MODE" FLAG (0=OFF, 1=ON)
 40     B=49408
 50     DATA PA1,PA2,PA3,PA4,PA5,OY,AY,EH,KK3,PP,JH,NN1,
        IH,TT2,RR1,AX,MM,TT1,DH1,IY,EY,DD1,UW1
 51     DATA AO,AA,YY2,AE,HH1,BB1,TH,UH,UW2,AW,DD2,GG3,
        VV,GG1,SH,ZH,RR2,FF,KK2,KK1,ZZ,NG,LL
 52     DATA WW,XR,WH,YY1,CH,ER1,ER2,OW,DH2,SS,NN2,HH2,
        OR,AR,YR,GG2,EL,BB2
 55     FOR X=1 TO 64
 60     READ NA$(X)
 70     NEXT X
 75     GOTO 450    : REM PROGRAM START UP TALK
 98     REM ***MAIN INPUT/DECODE LOOP***
100     GOSUB 400   : REM DISPLAY CURRENT STRING
101     IF IN=1 THEN PRINT "**Insert Mode**"
103     I$="b" : J=1
105     INPUT I$
106     IF I$=""  GOTO 300 : REM TALK
108     IF I$="E" THEN END : REM EXIT
109     IF I$="L" GOTO 115 : REM MOVE LEFT
110     IF I$="R" GOTO 121 : REM MOVE RIGHT
111     IF I$="D" GOTO 131 : REM DELETE PHONEME
112     IF I$="I" GOTO 140 : REM INSERT PHONEME
113     IF ASC(I$)<58 THEN 1000
114     GOTO 150
115     REM ***MOVE LEFT***
116     IF P-J<1 GOTO 165
117     P=P-J
119     GOTO 100
121     REM ***MOVE RIGHT***
123     IF P+J>L GOTO 165
125     IF P+J>64 GOTO 165
126     P=P+J
129     GOTO 100
131     REM ***DELETE PHONEME***
132     IF P>L GOTO 100
133     FOR I=P TO L
134     PH$(I)=PH$(I+1)
135     PH(I)=PH(I+1)
136     NEXT I
```

```
137     L=L-1
139     GOTO 100
140     REM ***TURN ON "INSERT MODE"***
145     IN=ABS(IN-1)
147     GOTO 100
150     REM ***TEST FOR VALID PHONEME ENTRY***
152     FOR X=1 TO 64
155     IF I$=NA$(X) GOTO 190
160     NEXT X
165     REM ***USER ENTERED INVALID DATA***
168     PRINT "***INVALID ENTRY***"
170     GOTO 103 : REM RETURN TO MAINLINE,
        DON'T REFRESH SCREEN
190     REM ***MODIFY STRING***
195     X=X-1
200     IF P>64 GOTO 165
201     IF IN=0 GOTO 210 : REM TEST FOR "INSERT MODE"
        BEING ON
202     IF L>63 GOTO 165 : REM ERROR-NO ROOM TO INSERT
203     FOR J=L+1 TO P+1 STEP-1
204     PH$(J) = PH$(J-1)
205     PH(J) = PH(J-1)
206     NEXT J
208     L=L+1
210     PH$(P)=I$
215     PH(P)=X
216     IF L<P THEN L=P
217     P=P+1
225     GOTO 100
300     REM ***TALK***
305     IF L<1 GOTO 100
320     FOR J=1 TO L
325     Z=B+PH(J)
340     IF PEEK (B)<128 GOTO 340
350     POKE Z,0
355     NEXT J
360     IF PEEK (B)<128 GOTO 360
365     POKE B,0
367     IF FIRST = 1 THEN GOTO 490
370     GOTO 100
400     REM ***BUILD AND DISPLAY PHONEME STRING***
403     S$=""
405     IF P=1 THEN S$="->"
408     FOR X = 1 to L
410     S$ = S$ + PH$(X) + "b"
415     IF X = P-1 THEN S$ = S$ + "->"
420     NEXT X
425     HOME
430     VTAB 10: PRINT S$
435     RETURN
440     END
450     DATA 27,7,45,53,4,4,4,6,2,42,26,11,2,36,12,
             35,2,29,19,2,9,32,51,2,15,15,35,2
451     DATA 55,0,9,19,0,50,2,13,31,2,25,58,2,42,
             15,16,0,9,49,22,0,13,51,4
453     RL=52
455     FOR X = 1 TO RL
457     READ PH(X)
```

```
 459      NEXT X
 460      L = RL
 470      FIRST = 1
 480      GOTO 300
 490      L = 1: FIRST = 0
 500      GOTO 100
1000      LET J=VAL(I$)
1010      IF RIGHT$(I$,1) = "R" THEN 121
1020      IF RIGHT$(I$,1) = "L" THEN 115
1030      GOTO 165
```

SAMPLE PROGRAM

The following program describes how to add "N" phrases to your existing program.

Data Statements

These statements must appear in the program before the lines that enable the synthesizer to speak.

> NOTE: The decimal codes for each particular allophone (as shown in Table 9-1) must be inserted in the places denoted by an asterisk; 128 must be added to the *last* allophone code in each data statement.

```
 10   DIM A(10,255): REM 10 represents the number of phrases
      to be spoken.  255 represents the number of allophones in
      the longest phrase.
100   FOR N=1 TO 10: REM 10 equals the number of phrases.
110   X=1
120   READ A
130   A(N,X)=A
140   X=X+1
150   IF A<128 THEN 120
160   NEXT N
200   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      first phrase.
210   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      second phrase.
```

```
220   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      third phrase.
230   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      fourth phrase.
240   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      fifth phrase.
250   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      sixth phrase.
260   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      seventh phrase.
270   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      eighth phrase.
280   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      ninth phrase.
290   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      tenth phrase.
```

Subroutine

```
10000   B=-16128 (for Apple II, slot 1)
        B=49408 (for Apple IIplus and IIe, slot 1)
10010   X=1
10020   IF PEEK (B)<128 THEN 10020
10030   C=A(N,X): IF C>128 THEN C=C-128
10035   POKE B+C,0
10040   X=X+1
10050   IF A(N,X-1)<128 THEN 10020
10060   IF PEEK (B)<128 THEN 10060
10070   POKE B,0
10080   RETURN
```

First Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the first phrase.

```
1000   LET N=1
1010   GOSUB 10000
```

Second Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the second phrase.

```
2000    LET N=2
2010    GOSUB 10000
```

Third Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the third phrase.

```
3000    LET N=3
3010    GOSUB 10000
```

Nth Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the Nth phrase.

```
N000    LET N=N
N010    GOSUB 10000
```

LET'S GET TECHNICAL . . .

DRIVER SUBROUTINES

This next section describes the driver, or talk subroutines, for the Apple II, IIplus, and IIe. These driver subroutines can be used directly in your own application programs, or studied as examples.

Apple II

```
 12    B=-16128
710    FOR X=1 TO L
720    Z=B+PH(X)-1
730    IF PEEK(B)<128 THEN GOTO 730
740    POKE Z,0
```

```
750    NEXT X
760    IF PEEK(B)<128 THEN GOTO 760
770    POKE B,0
790    GOTO 100
```

Explanation of Program

Line 12 B=−16128

This sets the base address for slot 1.

Line 710 FOR X=1 TO L

L represents the length of the phoneme string to be spoken. X represents the loop counter.

Line 720 Z=B+PH(X)−1

The phoneme address of the speech memory (see Table 9-1) is added to the base address.

NOTE: The program addresses and the actual chip addresses are offset by 1. Therefore, we must subtract 1 from PH(X).

Line 730 IF PEEK(B)<128 THEN GOTO 730

If the load request is high causing D7 to be low (<128), the speech synthesizer is still talking and cannot accept the next address (see Load Request in the hardware section of Chapter Nine).

Line 740 POKE Z,0

After the load request goes low causing D7 to be high, this statement outputs Z (from line 325) on the address lines of the speech synthesizer and strobes the address load (see Address Load in the hardware section of Chapter Nine). This causes the speech synthesizer to talk. The load request goes high again, indicating that the speech synthesizer is talking.

Line 750 NEXT X

When the load request goes low, the next allophone may be latched on the address lines of the speech synthesizer.

Line 760 IF PEEK(B)<128 THEN GOTO 760

When J=L, this statement is executed and again it reads the

D7 line and waits for it to go high before outputting the next statement.

Line 770 POKE B,0

This is the base address which begins at 0 in Table 9-1. This will output a pause and end the allophone phrase.

Line 790 GOTO 100

Returns to basic program.

Apple IIplus and IIe

```
 40   B=49408
300   REM ***TALK***
305   IF L<1 GOTO 100
320   FOR J=1 TO L
325   Z=B+PH(J)
340   IF PEEK(B)<128 GOTO 340
350   POKE Z,0
355   NEXT J
360   IF PEEK(B)<128 GOTO 360
365   POKE B,0
367   IF FIRST=1 THEN GOTO 490
370   GOTO 100
```

Explanation of the Program

Line 40 B=49408

This sets the base address for slot 1. You must add 256 to this number for each additional slot.

Line 305 IF L<1 GOTO 100

L<1 is an illegal length of string. If L<1 the system will require a different input.

Line 320 FOR J=1 TO L

L represents the length of the phoneme string to be spoken. J represents the loop counter.

Line 325 Z=B+PH(J)

The phoneme address of the speech memory (see Table 9-1) PH(J) is added to the base address.

Line 340 IF PEEK(B)<128 GOTO 340

If the load request is high causing D7 to be low (<128), the speech synthesizer is still talking and cannot accept the next address (see Load Request in the hardware section of Chapter Nine).

Line 350 POKE Z,0

After the load request goes low causing D7 to be high, this statement outputs Z (from line 325) on the address lines of the speech synthesizer and strobes the address load (see Address Load in the hardware section of Chapter Nine). This causes the speech synthesizer to talk. The load request again goes high indicating that the speech synthesizer is talking.

Line 355 NEXT J

When the load request goes low, the next allophone may be latched on the address lines of the speech synthesizer.

Line 360 IF PEEK(B)<128 GOTO 360

When J=L, this statement is executed and again it reads the D7 line and waits for it to go high before outputing the next statement.

Line 365 POKE B,0

This is the base address which begins at 0 in Table 9-1. This will output a pause and end the allophone phrase.

Line 367 IF FIRST=1 THEN GOTO 490

Due to the message spoken at program start-up, L was set equal to 52. This statement sets L back to 1.

Line 370 GOTO 100

The system is ready to edit the existing string if desired.

DEBUGGING YOUR CIRCUIT

The following steps are simple test procedures to follow when debugging your speech synthesis circuit. Please note that additional test equipment may be required (e.g., oscilloscope, voltmeter, logic probe). In addition to the test equipment, you should be experienced with debugging electrical circuits.

Procedures to Ensure That Your Connections Are Correct

1. Check all the power and ground connections. Pins 7, 19, and 23 of the SPO256-AL2 are +5 V. Pins 2 and 22 of the SPO256-AL2 are negatively grounded.
2. Disconnect the audio filter circuit from pin 24. Turn up the volume on the LM386 to maximum. Send another audio signal through the audio filter circuit. The output of a transistor radio is sufficient. Tune into a station where speech is playing rather than music. If the signal is heard at the speaker, this circuit is OK (it may be distorted somewhat). When resetting the chip, pin 24 should go to a logic 0 level. When the device is not speaking, a 40-kHz square wave is present on this pin.
3. Pins 2 and 25 of the SPO256-AL2 should be a logic 1 (+5 V). Pressing the reset button should change it to a logic 0 (0 V) and a small click should be heard in the speaker. If this doesn't occur and Procedures 1 and 2 are OK, the problem has to be in the 100-kΩ resistor, the push button switch, or the SPO256-AL2.
4. The signal on pin 26 should be oscillating. If this does not occur, either the crystal is bad or the capacitors connected to ground are the wrong capacitance. An oscilloscope or logic probe is required for this test.
5. Disconnect all the address lines (pins 10, 11, 13–18) from your microcomputer and then ground the pins (except pin

15). Connect pin 15 to +5 V. Disconnect the $\overline{\text{ALD}}$ (pin 20) and connect a momentary switch between it and +5 V. Upon depressing the momentary switch, allophone KK3 should be heard. If Steps 1–4 are operating properly and this step fails, the SPO256-AL2 is malfunctioning.

6 If Steps 1–5 are operating, proceed with the following: For the Apple II, enter POKE −16123,0. For the Apple IIplus and IIe, enter POKE 49413,0. Immediately on entering these statements, the state of the $\overline{\text{LRQ}}$ line (pin 9 of the SPO256-AL2) will change from a logic 0 to a logic 1. If this does not occur, then carefully recheck all your wiring and try substituting new TTL logic parts (i.e., 74LS00, 74LS02, 74LS368).

NOTE: An oscilloscope is required to debug the logic parts. Since these parts are so inexpensive, I suggest you replace them before testing them.

CHAPTER FIVE

MAKING YOUR TRS-80 MODEL I TALK

Using the Expansion Interface

The TRS-80 Model I computer was one of the first computers designed by Radio Shack. It is a very useful device that has been supported by Radio Shack and third-party software vendors for a number of years. Its large software base makes it an attractive computer with which to add speech—this is now possible and affordable and this chapter will show you how.

This chapter consists of two sections. One describes the hardware interfaces, schematics, and parts required to make your TRS-80 computer talk. The other explains the software required to drive the associated hardware. A detailed discussion of the program listing and its editing features is also included.

HARDWARE

This section explains all that's required to build a speech interface unit for your TRS-80 Model I computer, using the expansion interface. It is divided into three parts: the *Starter Kit,* the *Complete Kit,* and the *Complete Module.*

The starter kits for each particular model contain the components denoted in Table 5-1 by an asterisk. (They usually include the speech synthesis chip, the ceramic resonator, and the TTL components.) The remaining components (experimenter's breadboard, connector, ribbon cable, resistors, capacitors, etc.) have to be purchased separately. These parts are very popular and if you don't have them around your lab, any electronics store (e.g., Radio Shack) will surely have them. The starter kit is the least expensive type

Table 5-1 TRS-80 MODEL I EXPANSION INTERFACE PARTS LIST

QUANTITY	PC BOARD IDENTIFICATION FOR THE COMPLETE KIT	DESCRIPTION
1	Not applicable for this computer	PC board (3″ × 4″) or breadboard
1		Reset switch
1		Power jack (mini)
1		RCA phono speaker jack (¼″)
1		10K potentiometer (volume control, 3-prong)
1		40-pin connector (female)
1		7805 C or LM340-5 5-V regulator
1		* SPO256-AL2 speech synthesizer
1		* 74LS27 TTL logic circuit
1		* 74LS30 TTL logic circuit
1		* 74LS368 TTL logic circuit
1		* 74LS374 TTL logic circuit
1		LM386–1 operational amplifier (This is required to drive a 4- or 8-Ω speaker.)
2		14-pin sockets (for the 74LS27 and 74LS30)
1		20-pin socket (for the 74LS374)
1		16-pin socket (for the 74LS368)
1		8-pin socket (for the LM386)
1		28-pin socket (for the SPO256-AL2)
1		Ribbon cable—the length of the cable is optional.
1		SPST on/off power switch

QUANTITY	PC BOARD IDENTIFICATION FOR THE COMPLETE KIT	DESCRIPTION
1		* 3.12-MHz ceramic resonator (blue, CSA3.12MS2)
4 Resistors:		
2		33-kΩ (orange, orange, orange)
1		10-kΩ (brown, black, orange) or 100-kΩ (brown, black, yellow)
1		10-Ω (brown, black, black) or 11-Ω (brown, brown, black)
12 Capacitors:		
3		0.1 μF
1		100-μF electrolytic (audio filter)
1		10-μF electrolytic (power filter)
1		1-μF electrolytic
1		10-μF electrolytic (audio filter)
1		100-μF electrolytic (power filter)
2		0.025 μF
2		100 pF
Additional parts required (*not included*)		
1		7.5 to 9-V dc power supply (250–300mA). Can be obtained at Radio Shack, Catalog #2731455 or Catalog #603053.
1		Speaker—any 4- or 8-Ω speaker may be used; the choice is yours.

* The only parts included in the starter kits.

of kit; however, it requires the most amount of time to build the circuit. If you are not familiar with wiring up breadboards, you may want to try the complete kit or complete module. The starter kits were designed for the more experienced hobbyist who wants to save a little extra money in exchange for a portion of his or her time.

The complete kits consist of all the parts listed in Table 5-1, including printed circuit (PC) boards, connectors, and cabinet. The speaker and the power supply are the only parts that have to be supplied. This kit is somewhat more expensive, but offers the ease of building the circuit on a PC board specifically designed for your particular computer. No wiring experience is necessary. All you need to know is how to use a soldering iron. This circuit is designed for the beginner who wants to learn how to build his or her own circuit and at the same time save a little money.

If after reading the instructions you are not interested in building a synthesizer yourself, you can purchase a complete module specifically designed for your computer. The complete modules are fully assembled and tested, and plug directly into the various computers, enabling them to speak in a matter of minutes!

> NOTE: The parts that form the starter kits, complete kits, and complete modules can be purchased from a number of distributors listed in the Parts Supplier Listing, Appendix B. With some distributors you may have to purchase all the parts separately, with others you may be able to purchase the parts in kit or module form. (Refer to this listing and note the parts that each company supplies before setting out to purchase your speech synthesis kits.)

All circuits, kit or module form, conform to certain requirements; they are discussed below.

PORT LOCATIONS AND POWER SPECIFICATIONS

This board plugs into the 40-pin expansion bus of the TRS-80 Model I. The synthesizer board is decoded to ports 252 and 253. This

circuit does not provide internal power so the board *must* be powered externally.

PARTS LISTING

The list in Table 5-1 describes all parts required to build a speech synthesizer for your computer. Recall that the parts denoted by an asterisk combine to form the starter kits. All additional parts must be supplied and wired up on an experimenter's breadboard. The complete kits include all the parts in the list with the exception of a speaker and power supply. The PC board identification (included for complete kits only) describes where the parts should be placed on the PC board (see the section on Complete Kit Assembly Instructions).

STARTER KIT ASSEMBLY INSTRUCTIONS

When building the starter kit, refer to the schematic diagram in Fig. 5-1.

NOTE: After completing your starter kit, see the section on Complete Module and Operating Instructions.

STEP 1 In addition to the starter kit, obtain all of the components listed in the TRS-80 Model I Expansion Parts List, Table 5-1.

STEP 2 Insert the sockets into the breadboard and make a note of where pin 1 is to be located. (Pin 1 should be in the upper left-hand corner.)

STEP 3 Connect the ribbon cable to the 40-pin female connector. The connector states what the pin numbers are.

STEP 4 Solder or wirewrap the following connections:

FROM THE 40-PIN CONNECTOR:	*TO:*
GND pin 8	Common ground of the synthesizer board
A7 pin 36	Pin 1 of the 74LS30

FROM THE 40-PIN CONNECTOR:	*TO:*
A1 pin 27	Pin 4 of the 74LS368
A2 pin 40	Pin 3 of the 74LS30
A3 pin 34	Pin 4 of the 74LS30
A4 pin 31	Pins 5 and 12 of the 74LS30
A5 pin 35	Pin 6 of the 74LS30
A6 pin 38	Pin 11 of the 74LS30
A0 pin 25	Pins 2 and 10 of the 74LS27 and pin 10 of the 74LS368
OUT pin 12	Pins 13 and 3 of the 74LS27
IN pin 19	Pin 11 of the 74LS27
D0 pin 30	Pin 30 of the 74LS374
D1 pin 22	Pin 4 of the 74LS374
D2 pin 32	Pin 7 of the 74LS374
D3 pin 26	Pin 8 of the 74LS374
D4 pin 18	Pin 13 of the 74LS374
D5 pin 28	Pin 14 of the 74LS374
D6 pin 24	Pin 17 of the 74LS374
D7 pin 20	Pin 18 of the 74LS374 and pin 13 of the 74LS368

STEP 5 Next solder or wirewrap these connections:

FROM THE 74LS30:	*TO:*
Pin 2	Pin 5 of the 74LS368
Pin 8	Pins 1, 5, and 9 of the 74LS27

FROM THE 74LS27:	*TO:*
Pin 12	Pin 2 of the 74LS368
Pin 6	Pin 11 of the 74LS374
Pin 8	Pin 6 of the 74LS368
Pin 4	Pin 9 of the 74LS368

FROM THE 74LS368:	*TO:*
Pin 3	Pin 20 of the SPO256-AL2
Pin 7	Pin 15 of the 74LS368
Pin 14	Pin 9 of the SPO256-AL2

FROM THE 74LS374:	*TO:*
Pin 2	Pin 18 of the SPO256-AL2
Pin 5	Pin 17 of the SPO256-AL2
Pin 6	Pin 16 of the SPO256-AL2
Pin 9	Pin 15 of the SPO256-AL2
Pin 12	Pin 14 of the SPO256-AL2
Pin 15	Pin 13 of the SPO256-AL2
Pin 16	Pin 11 of the SPO256-AL2
Pin 19	Pin 10 of the SPO256-AL2

STEP 6 The Power and Ground Connections

FROM SIDE 2 OF THE SPST ON/OFF SWITCH	*TO:*
	Pin 14 of the 74LS30
	Pin 14 of the 74LS27
	Pins 12 and 16 of the 74LS368
	Pin 20 of the 74LS374
	Pins 7, 19, and 23 of the SPO256-AL2

FROM PIN 8 OF THE 40-PIN CONNECTOR	*TO:*
	Pin 7 of the 74LS30
	Pin 7 of the 74LS27
	Pins 1 and 8 of the 74LS368
	Pins 1 and 10 of the 74LS374
	Pins 1 and 22 of the SPO256-AL2

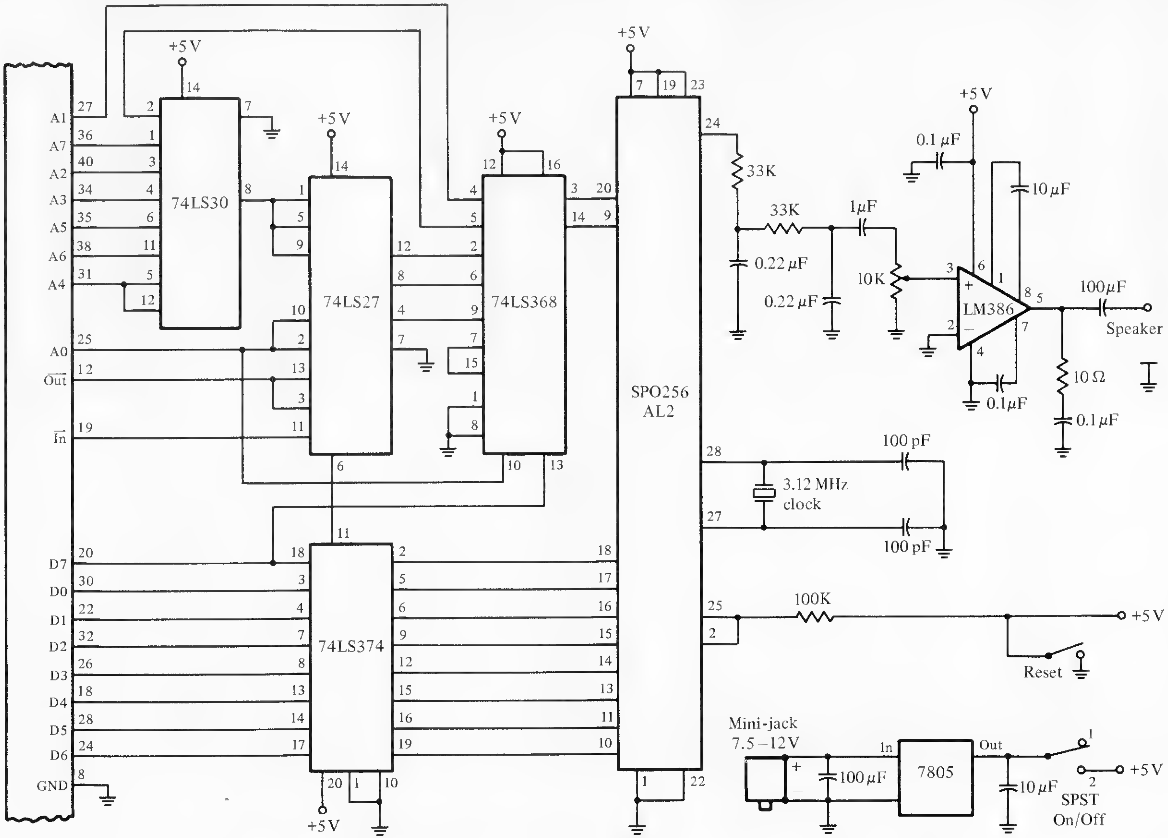


Fig. 5-1 TRS-80 Model I Expansion Interface speech circuit schematic.

STEP 7 The Reset Circuit

a. Connect a 100-kΩ resistor between +5 V (output of the 7805) and pins 2 and 25 of the SPO256-AL2.
b. Connect the reset switch between ground (pin 8 of the 40-pin connector) and pins 2 and 25 of the SPO256-AL2.

STEP 8 The Regulator Circuit

a. Connect the positive side of the power jack to the 9-V input side of the 7805 (or LM340).
b. Connect the negative side of the power jack to the negative terminal of the 7805 (or LM340).
c. Connect the negative terminal of the 7805 (or LM340) to pin 8 of the 40-pin connector.
d. Connect the 5-V output of the 7805 (or LM340) to side 1 of the SPST on/off switch.
e. Connect the positive side of a 100-μF electrolytic capacitor between the 9-V input of the 7805 (or LM340) and the negative side to ground (pin 8 of the 40-pin connector).
f. Connect the positive side of a 10-μF electrolytic capacitor between the 5-V output of the 7805 (or LM340) and the negative side to ground (pin 8 of the 40-pin connector).

STEP 9 The Oscillator Circuit

a. Connect the 3.12-MHz ceramic resonator between pins 27 and 28 of the SPO256-AL2.
b. Connect a 100-pF capacitor between pin 27 of the SPO256-AL2 and ground (pin 8 of the 40-pin connector).
c. Connect a 100-pF capacitor between pin 28 of the SPO256-AL2 and ground (pin 8 of the 40-pin connector).

STEP 10 The Audio Filter Circuit

a. Connect side 1 of a 33-kΩ resistor to pin 24 of the SPO256-AL2.
b. Connect side 2 of the first 33-kΩ resistor to side 1 of the second 33-kΩ resistor and to side 1 of the first

0.022-μF capacitor. Connect side 2 of the first 0.022-μF capacitor to ground.

c. Connect side 2 of the second 33-kΩ resistor to the positive side of the 1-μF electrolytic capacitor and to side 1 of the second 0.022-μF capacitor. Connect side 2 of the 0.022-μF capacitor to ground.

d. Connect the negative side of the 1-μF electrolytic capacitor to side 1 of the 10K potentiometer.

e. Connect side 2 (middle terminal) of the 10K potentiometer to pin 3 of the LM386.

f. Connect side 3 of the 10K potentiometer to ground.

g. Connect pins 2 and 4 of the LM386 to ground.

h. Connect a 0.1-μF capacitor between pins 4 and 7 of the LM386.

i. Connect the positive side of the 10-μF electrolytic capacitor to pin 1 of the LM386 and the negative side to pin 8 of the LM386.

j. Connect pin 6 of the LM386 to +5 V.

k. Connect a 0.1-μF capacitor between pin 6 of the LM386 and ground.

l. Connect pin 5 of the LM386 to side 1 of a 10-Ω resistor. Connect side 2 of the 10-Ω resistor to side 1 of a 0.1-μF capacitor. Connect side 2 of the 0.1-μF capacitor to ground.

m. Connect the positive side of the 100-μF electrolytic capacitor to pin 5 of the LM386. Connect the negative side of the 100-μF capacitor to side 1 of the speaker jack. Connect side 2 of the speaker jack to ground.

STEP 11 Insertion of the Integrated Circuits

NOTE: When inserting the IC's, be sure that pin 1 of the IC lines up with pin 1 of the socket that was labeled in step 2.

a. Insert the SPO256-AL2 into the 28-pin socket.

b. Insert the 74LS30 into the 14-pin socket (the socket connected to lines A1–A7 of the 40-pin connector).

c. Insert the 74LS27 into the 14-pin socket (the socket connected to A0, $\overline{\text{OUT}}$, and $\overline{\text{IN}}$ of the 40-pin connector).

d. Insert the 74LS368 into the 16-pin socket.

e. Insert the LM386 into the 8-pin socket.
f. Insert the 74LS374 into the 20-pin socket.

STEP 12 Proceed to the section on the Complete Module and Operating Instructions.

COMPLETE KIT ASSEMBLY INSTRUCTIONS

Unfortunately, a PC board and complete module has not been designed for this interface. Please refer to Chapter 6.

COMPLETE MODULE AND OPERATING INSTRUCTIONS

Once you have built the starter kit, you are now ready to operate your speech synthesizer (see Fig. 5-1).

Unfortunately, a PC board and complete module was not designed for the TRS-80 Model I Expansion Interface. The following applies to the starter kit circuit only.

Operating Procedure

STEP 1 Switch the on/off switch to the off position.

STEP 2 Plug the speaker into the speaker jack.

STEP 3 Plug the external power supply into the power jack of the synthesizer breadboard (the tip is positive). External power is required at all times for this circuit.

NOTE: Do not plug into wall outlet yet.

CAUTION: The computer must be off.

STEP 4 Plug the synthesizer into the 40-pin bus of the TRS-80 Model I.

STEP 5 Plug in the 9-V power supply to the synthesizer into the wall outlet.

STEP 6 Turn your computer on.

STEP 7 Turn the synthesizer on/off switch to the on position.
STEP 8 Depress the reset button.

NOTE: A faint "click" should be heard in the speaker when depressing this button. If this sound is not heard, turn up the volume and try it again. If this condition still exists, proceed to the following section.

TESTING YOUR CIRCUIT

After carefully wiring up your speech synthesizer board, the following commands can be used to test your circuit before loading in the respective programs. Before power is applied, visually inspect your hardware to ensure that the proper connections have been made and all the grounds are secure. On power up, a hardware reset is required—simply close the switch momentarily. A click or pop should be heard in the speaker. If this occurs, proceed with the instructions in this section. If this condition does not occur, refer to the section on Debugging Your Hardware at the end of this chapter.

To supply the proper address to the speech synthesizer, port 253 is used. The $\overline{\text{ALD}}$ latch is port 252. A simple OUT 253,5 will output address 5 (the OY allophone) on the data bus of the speech synthesizer. Then OUT 252,0 will strobe $\overline{\text{ALD}}$, causing the synthesizer to talk. If everything is correct, the synthesizer will continue to speak until a pause is entered. To enter a pause and silence the board, simply enter: OUT 253,0. This is a pause at location 000. Then type the statement OUT 252,0. This is the $\overline{\text{ALD}}$ strobe. You are now ready to load in the Exclusive Program in the usual manner (as specified in your computer manual) and create your own allophone phrases.

SOFTWARE

The Exclusive Phrase Finder Program that follows allows you to build words and phrases from their constituent allophones. The

phrase can be edited by moving a pointer left or right to the desired position (see Table 5-2). Inserting, deleting, or replacing allophones can then be accomplished quite easily. When the phrase is prepared to your satisfaction, a simple ENTER, NEW LINE, or RETURN will signal the synthesizer to talk.

Table 5-2 ALLOPHONE EDITING COMMANDS

AVAILABLE COMMANDS	OPERATION
"phoneme strings"	Causes named allophone to be added to the phrase at the current position of the pointer, by either replacing the existing allophone or inserting one before it (see the section on Inserting an Allophone).
"L"	Moves the position pointer left one allophone.
"R"	Moves the position pointer right one allophone.
"D"	Deletes allophone at the current position pointer.
"XL"	Moves the position pointer "X" number of allophones to the left.
"XR"	Moves the position pointer "X" number of allophones to the right.
"I"	Turns on the "insert mode." The next allophone entered will be inserted into the phrase at the current position of the pointer. Additional allophones will be inserted until "I" is entered again. The second "I" command will turn the insert mode off. When the insert mode is off, an entered allophone will replace the one at the current position. This is the default at system start-up.
NEW LINE, ENTER, OR RETURN	Causes the system to output to the hardware the commands necessary to pronounce the phase.
"E"	Exits the program.

EXCLUSIVE PHRASE FINDER PROGRAM DESCRIPTION

At system start-up, a brief message will be spoken and the following commands are performed.

> NOTE: This message will be spoken each time the program is RUN. To delete this message from the system, the command "Delete line 75 from the program" must be issued.

The screen is cleared and all variables are initialized. The allophone symbol array is initialized with the 64 two- or three-character symbols that represent each allophone. These are strings the user will enter in order to add an allophone to the phrase (see Tables 3-1 and 9-1). The position pointer is at position one, and the user is prompted for input with a "⩾." At this prompt, an allophone or any of the commands in Table 5-2 may be entered. An invalid allophone will be flagged as an error, as will attempting to move the position pointer to the left or right of the boundaries of the phrase. After each command, the updated phrase is displayed with the current position indicated by "⩾."

For example, at system start-up, the screen will look like this:

COMMAND	*SCREEN*	*COMMENTS*
RUN	⩾ ?	Pointer at position one.
HH1	⩾ ?HH1	Desired allophone.
ENTER	HH1⩾ ?	The first allophone has been entered; the pointer is at position two; the system is ready for the next input.
EH1	HH1⩾ ?EH1	Next allophone.

COMMAND	*SCREEN*	*COMMENTS*
ENTER	HH1> ?EH1 ***INVALID ENTRY***	User entered invalid data.
EH	HH1> ?EH	
ENTER	HH1 EH>	The second allophone has been entered; the pointer is at position three; the system is ready for the next input.

Editing Features

Upon entering a string of allophones and noticing that a few corrections are in order, the following edit commands are useful.

First, we must position the pointer at the location where an editing command is to be performed. Let's take the word "hello" for example. The screen should now look like this:

HH1 EH LL UW1>

Realizing that the UW1 allophone is incorrect, we would like to REPLACE it with "OW." To do so, the following commands are required:

Replacing an Allophone

COMMAND	*SCREEN*	*COMMENTS*
"L"	HH1 EH LL UW1> L	We have to move the pointer left one space to replace UW1.

COMMAND	*SCREEN*	*COMMENTS*
"ENTER"	HH1 EH LL≥ UW1	The system is now ready to replace the allophone UW1.
"OW"	HH1 EH LL≥ UW1 OW	The desired replacement allophone is typed.
"ENTER"	HH1 EH LL OW≥	After pressing ENTER, the allophone has been replaced.

Moving the position pointer right works in the same manner as moving to the left. The only exception is that we use an "R" instead of an "L." These commands move the pointer one space at the time.

In replace mode (the default at program start-up), the new allophone will replace the allophone at the current position in the phrase.

NOTE: If you attempt to move the position pointer left or right beyond its boundaries, the message ***INVALID ENTRY*** will appear.

COMMAND	*SCREEN*	*COMMENTS*
"R" ENTER	HH1 EH LL OW≥ R ***INVALID ENTRY***	To clear the invalid entry, press ENTER or type in a valid command.
"L" ENTER	≥HH1 EH LL OW L ***INVALID ENTRY***	

The XL and XR commands work in the same manner as the L and R commands. The only difference is that these commands move the position pointer "X" number of allophones (or spaces)

to the left or right. Once again, exceeding the boundaries will prompt an ***INVALID ENTRY*** message. For example:

COMMAND	*SCREEN*	*COMMENTS*
"2L"	HH1 EH LL OW1>	
ENTER	HH1 EH> LL OW	The position pointer has moved 2 spaces to the left.

Deleting an Allophone. We'll use the same examples as above. Remember, we must first position the pointer to the specified allophone to be deleted. Once this is accomplished, the following commands are required.

COMMAND	*SCREEN*	*COMMENTS*
"L" ENTER (3 times)	HH1 EH EH LL OW>	We need to delete an EH here, so first we must move the pointer 3 spaces to the left.
"D"	HH1 EH> EH LL OW D	The pointer is now positioned to the allophone to be deleted.
"ENTER"	HH1 EH> LL OW	The EH allophone has been deleted.

NOTE: The delete command "D" deletes one allophone at a time.

Inserting an Allophone. After creating your allophone phrases, and realizing that a few pauses (or other allophones) need to be inserted, the following command sequence must be performed. Let's use the word "chatter" as an example.

COMMAND	*SCREEN*	*COMMENTS*
"L" ENTER (3 times)	CH AE TT2 ER1 PA3> ?	We must position the pointer at the location where the inserted allophone will go.
"I"	CH AE> TT2 ER1 PA3 ?I	We are ready to turn on the "insert mode."
"ENTER"	CH AE> TT2 ER1 PA3 ?	Insert mode has been turned on.
"PA3"	CH AE> TT2 ER1 PA3 ? PA3	The desired allophone to be inserted is typed.
"ENTER"	CH AE PA3> TT2 ER2 PA3 ?	The allophone has been inserted. Additional allophones may be entered at this time if required.
"I"	CH AE PA3> TT2 ER2 PA3 ?I	We are ready to turn off the insert mode.
"ENTER"	CH AE PA3> TT2 ER2 PA3 ?	Insert mode has been turned off.

MAKING YOUR COMPUTER TALK

A simple "ENTER," "RETURN," or "NEW LINE" is all that's required to make the system talk. If a new allophone string is desired, you must first EXIT (E) the program and then RUN it again. This will clear all the previous allophone codes stored. Unfortunately at the time of writing this book, a "SAVE" routine was not supplied with the Exclusive Phrase Finder Program. You

may want to try writing your own subroutine to save the allophone strings you've created with the program (in separate files). An alternative method is to save the Exclusive Phrase Finder Program on cassette tape or disk, with the phrase or word you've just created. When reloading the program, the following command is necessary for it to speak the prestored phrase.

GOTO 300

Upon typing this command, the computer will speak the prestored allophone string and the allophone codes will be printed on the screen. The system is now ready to edit the existing string. To enter a new string, RUN the program.

EXCLUSIVE PHRASE FINDER PROGRAM LISTING

TRS-80 Model I Expansion Interface

NOTE: When typing in the allophones, it is essential to type a blank space in place of all **b**'s in the program below.

```
 0      CLS
 2      PRINT "***PHONEME CONCATENATION SYSTEM***"
 5      CLEAR 1000
10      DIM NA$(64) : REM PHONEME SYMBOL ARRAY
12      DIM PH$(64) : REM CURRENT PHONEME STRING
14      DIM PH(64)  : REM CURRENT PHONEME STRING ADDRESSES
16      S$ = ""     : REM DISPLAY STRING, WITH CURSOR POSITION
18      I$ = ""     : REM INPUT STRING FROM USER
25      P=1         : REM CURRENT POSITION POINTER
30      L=0         : REM LENGTH OF PHONEME STRING
35      IN=0        : REM "INSERT MODE" FLAG (0=OFF, 1=ON)
50      DATA PA1,PA2,PA3,PA4,PA5,OY,AY,EH,KK3,PP,JH,
        NN1,IH,TT2,RR1,AX,MM,TT1,DH1,IY,EY,DD1,UW1
51      DATA AO,AA,YY2,AE,HH1,BB1,TH,UH,UW2,AW,DD2,
        GG3,VV,GG1,SH,ZH,RR2,FF,KK2,KK1,ZZ,NG,LL
52      DATA WW,XR,WH,YY1,CH,ER1,ER2,OW,DH2,SS,NN2,
        HH2,OR,AR,YR,GG2,EL,BB2
55      FOR X=1 TO 64
60      READ NA$(X)
70      NEXT X
75      GOTO 450    : REM PROGRAM START UP TALK
```

```
 98  REM ***MAIN INPUT/DECODE LOOP***
100  GOSUB 400   : REM DISPLAY CURRENT STRING
101  IF IN=1 THEN PRINT "**Insert Mode**"
103  I$="" : J=1
105  INPUT I$
106  IF I$=""  GOTO 300 : REM TALK
108  IF I$="E" THEN END : REM EXIT
109  IF I$="L" GOTO 115 : REM MOVE LEFT
110  IF I$="R" GOTO 121 : REM MOVE RIGHT
111  IF I$="D" GOTO 131 : REM DELETE PHONEME
112  IF I$="I" GOTO 140 : REM INSERT PHONEME
113  IF ASC(I$)<58 THEN 1000
114  GOTO 150
115  REM ***MOVE LEFT***
116  IF P-J=0 GOTO 165
117  P=P-J
119  GOTO 100
121  REM ***MOVE RIGHT***
123  IF P+J>L+1 GOTO 165
125  IF P+J>64 GOTO 165
126  P=P+J
129  GOTO 100
131  REM ***DELETE PHONEME***
132  IF P>L GOTO 100
133  FOR I=P TO L
134  PH$(I)=PH$(I+1)
135  PH(I)=PH(I+1)
136  NEXT I
137  L=L-1
139  GOTO 100
140  REM ***TURN ON "INSERT MODE"***
145  IN=ABS(IN-1)
147  GOTO 100
150  REM ***TEST FOR VALID PHONEME ENTRY***
152  FOR X=1 TO 64
155  IF I$=NA$(X) GOTO 190
160  NEXT X
165  REM ***USER ENTERED INVALID DATA***
168  PRINT "***INVALID ENTRY***"
170  GOTO 103 : REM RETURN TO MAINLINE,
     DON'T REFRESH SCREEN
190  REM ***MODIFY STRING***
195  X=X-1
200  IF P>64 GOTO 165
201  IF IN=0 GOTO 210 : REM TEST FOR "INSERT MODE"
     BEING ON
202  IF L>63 GOTO 165 : REM ERROR-NO ROOM TO INSERT
203  FOR J=L+1 TO P+1 STEP-1
204  PH$(J) = PH$(J-1)
205  PH(J) = PH(J-1)
206  NEXT J
208  L=L+1
210  PH$(P)=I$
215  PH(P)=X
216  IF L<P THEN L=P
217  P=P+1
225  GOTO 100
300  REM ***TALK***
```

```
305     IF L<1 GOTO 100
320     FOR J=1 TO L
330     OUT 253, PH(J)
340     OUT 252, 0
350     IF INP(252)<128 GOTO 350
360     NEXT J
370     OUT 253, 0
380     OUT 252, 0
385     IF FIRST = 1 THEN GOTO 490
390     GOTO 100
400     REM ***BUILD AND DISPLAY PHONEME STRING***
403     S$=""
405     IF P=1 THEN S$="→"
408     FOR X = 1 TO L
410     S$ = S$ + PH$(X) + "b"
415     IF X = P-1 THEN S$ = S$ + "→"
420     NEXT X
425     CLS
430     PRINT a 256,S$
435     RETURN
440     END
450     DATA 27,7,45,53,4,4,4,6,2,42,26,11,2,36,12,
             35,2,29,19,2,9,32,51,2,15,15,35,2
451     DATA 55,0,9,19,0,50,2,13,31,2,25,58,2,42,
             15,16,0,9,49,22;0,13,51,4
453     RL=52
455     FOR X = 1 TO RL
457     READ PH(X)
459     NEXT X
460     L = RL
470     FIRST = 1
480     GOTO 300
490     L = 1: FIRST = 0
500     GOTO 100
1000    LET J=VAL(I$)
1010    IF RIGHT$(I$,1) = "R" THEN 121
1020    IF RIGHT$(I$,1) = "L" THEN 115
1030    GOTO 165
```

SAMPLE PROGRAM

The following program describes how to add "N" phrases to your existing program.

> NOTE: The decimal codes for each particular allophone (as shown in Table 9-1) must be inserted in the places denoted by an asterisk; 128 must be added to the last allophone code in each data statement.

Data Statements

These statements must appear in the program before the lines that enable the synthesizer to speak.

```
10   DEFINT A,N,X
20   DIM A(10,255): REM 10 represents the number of phrases
     to be spoken.  255 represents number of allophones in the
     longest phrase.
100  FOR N=1 TO 10: REM 10 equals the number of phrases.
110  X=1
120  READ A
130  A(N,X)=A
140  X=X+1
150  IF (A and 128)=0 THEN 120
160  NEXT N
200  DATA *,*,*, . . . ,*: REM decimal codes of allophones for
     first phrase.
210  DATA *,*,*, . . . ,*: REM decimal codes of allophones for
     second phrase.
220  DATA *,*,*, . . . ,*: REM decimal codes of allophones for
     third phrase.
230  DATA *,*,*, . . . ,*: REM decimal codes of allophones for
     fourth phrase.
240  DATA *,*,*, . . . ,*: REM decimal codes of allophones for
     fifth phrase.
250  DATA *,*,*, . . . ,*: REM decimal codes of allophones for
     sixth phrase.
260  DATA *,*,*, . . . ,*: REM decimal codes of allophones for
     seventh phrase.
270  DATA *,*,*, . . . ,*: REM decimal codes of allophones for
     eighth phrase.
280  DATA *,*,*, . . . ,*: REM decimal codes of allophones for
     ninth phrase.
290  DATA *,*,*, . . . ,*: REM decimal codes of allophones for
     tenth phrase.
```

Subroutine

```
1010    X=1
1020    IF INP(252)<128 GOTO 1020
1030    OUT 253, A(N,X)
1040    OUT 252,0
1070    IF (A(N,X) AND 128)=0
        THEN X=X+1: GOTO 1020
1080    IF INP(252)<128 GOTO 1080
1090    OUT 253,0
1100    OUT 252,0
1110    RETURN
```

First Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the first phrase.

```
1000    LET N=1
1010    GOSUB 10000
```

Second Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the second phrase.

```
2000    LET N=2
2010    GOSUB 10000
```

Third Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the third phrase.

```
3000    LET N=3
3010    GOSUB 10000
```

Nth Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the Nth phrase.

```
N000    LET N=N
N010    GOSUB 10000
```

LET'S GET TECHNICAL . . .

DRIVER SUBROUTINES

This next section describes the driver, or talk subroutines, for the TRS-80 Model I Expansion Interface. This driver subroutine can be used directly in your own application programs, or studied as an example.

```
300    REM *** TALK ***
305    IF L<1 GOTO 100
320    FOR J=1 TO L
330    OUT 253, PH(J)
340    OUT 252,0
350    IF INP(252)<128 GOTO 350
360    NEXT J
370    OUT 253,0
380    OUT 252,0
385    IF FIRST=1 THEN GOTO 490
390    GOTO 100
```

Explanation of Program

Line 305 IF L<1 GOTO 100

L<1 is an illegal length of string. If L<1, the system will require a different input.

Line 320 FOR J=1 TO L

L represents the length of the phoneme string to be spoken. J represents the loop counter.

Line 330 OUT 253, PH(J)

The phoneme address of the speech memory (see Table 9-1) is outputted on the address lines of the speech synthesizer to port 253.

Line 340 OUT 252,0

This strobes the address load causing the speech synthesizer to talk. (See Address Load in the hardware section of Chapter Nine.)

Line 350 IF INP(252)<128 GOTO 350

When the load request is high, causing D7 to be low (<128) the speech synthesizer is still talking and cannot accept the next address. (See Load Request in the hardware section of Chapter Nine.)

Line 360 NEXT J

When the load request goes low, causing D7 to go high, the next allophone may be latched on the address lines of the speech synthesizer.

Line 370 OUT 253,0

This will output a pause on the address lines of the speech synthesizer and end the allophone phrase.

Line 380 OUT 252,0

This strobes the address load again.

Line 385 IF FIRST=1 THEN GOTO 490

Due to the message spoken at program start-up, L was set equal to 52. This statement sets L back to 1.

Line 390 GOTO 100

The system is ready for the next input.

DEBUGGING YOUR CIRCUIT

The following steps are simple test procedures to follow when debugging your speech synthesis circuit. Please note that additional

test equipment may be required (e.g., oscilloscope, voltmeter, logic probe). In addition to the test equipment, you should be experienced with debugging electrical circuits.

Procedures to Ensure That Your Connections Are Correct

1. Check all the power and ground connections. Pins 7, 19, and 23 of the SPO256-AL2 are +5 V. Pins 2 and 22 of the SPO256-AL2 are negatively grounded.
2. Disconnect the audio filter circuit from pin 24. Turn up the volume on the LM386 to maximum. Send another audio signal through the audio filter circuit. The output of a transistor radio is sufficient. Tune into a station where speech is playing rather than music. If the signal is heard at the speaker, this circuit is OK (it may be distorted somewhat). When resetting the chip, pin 24 should go to a logic 0 level. When the device is not speaking, a 40-kHz square wave is present on this pin.
3. Pins 2 and 25 of the SPO256-AL2 should be a logic 1 (+5 V). Pressing the reset button should change it to a logic 0 (0 V) and a small click should be heard in the speaker. If this doesn't occur and Procedures 1 and 2 are OK, the problem has to be in the 100-kΩ resistor, the push button switch, or the SPO256-AL2.
4. The signal on pin 26 should be oscillating. If this does not occur, either the crystal is bad or the capacitors connected to ground are the wrong capacitance. An oscilloscope or logic probe is required for this test.
5. Disconnect all the address lines (pins 10,11,13–18) from your microcomputer and then ground the pins (except pin 15). Connect pin 15 to +5 V. Disconnect the $\overline{\text{ALD}}$ (pin 20) and connect a momentary switch between it and +5 V. Upon depressing the momentary switch, allophone KK3 should be heard. If Steps 1–4 are operating properly and this step fails, the SPO256-AL2 is malfunctioning.

6. If Steps 1–5 are operating, proceed with the following. Enter the statements $\overline{\text{OUT}}$ 253,5 and $\overline{\text{OUT}}$ 252,0. Immediately on entering $\overline{\text{OUT}}$ 252,0 the state of the $\overline{\text{LRQ}}$ line (pin 9 of the SPO256-AL2) will change from a logic 0 to a logic 1. If this does not occur, then carefully recheck all your wiring and try substituting new TTL logic parts (i.e., 74LS30, 74LS27, 74LS374, 74LS368).

NOTE: An oscilloscope is required to debug the logic parts. Since these parts are so inexpensive, I suggest you replace them before testing them.

CHAPTER SIX

MAKING YOUR TRS-80 MODELS I, III, AND IV TALK

Using the Printer Port Interface

The TRS-80 Model III and Model IV computers are upgrades of the TRS-80 Model I. Most of the software that was written for the Model I has also been upgraded to run on the Models III and IV. The addition of a speech unit makes your Model III or IV even more advanced—a computer that talks!

This chapter explains the hardware and software required to make your TRS-80 computer talk. It is divided into two sections: one includes the schematics and parts required for the speech circuit, while the other explains and provides the software required to drive the associated hardware. A detailed discussion of the program listing and its editing features are also included.

HARDWARE

This section explains all that's required to build a speech interface unit for your TRS-80 Models I, III, and IV computers, using the printer port interface. It is divided into three parts: the *Starter Kit,* the *Complete Kit,* and the *Complete Module.*

The starter kits for each particular model contain the components denoted in Table 6-1 by an asterisk. (They usually include the speech synthesis chip, the ceramic resonator, and the TTL components.) The remaining components (experimenter's breadboard, connector, ribbon cable, resistors, capacitors, etc.) have to be purchased separately. These parts are very popular and if you don't

Table 6-1 TRS-80 PRINTER PORT INTERFACE MODELS I, III, AND IV PARTS LIST

QUANTITY	PC BOARD IDENTIFICATION FOR THE COMPLETE KIT	DESCRIPTION
1		PC Board (3″ × 4″) or Breadboard
1	25	Reset switch
1	24	Power jack (mini)
1	23	RCA phono speaker jack ($\frac{1}{4}$″)
1	22	10K potentiometer (volume control, 3-prong)
1		34-pin connector (female)
1	20,21	7805 C or LM340–5 5-V regulator
1		* SPO256-AL2 speech synthesizer
1		LM386–1 operational amplifier (This is required to drive a 4- or 8-Ω speaker.)
1	2	8-pin socket (for the LM386)
1	1	28-pin socket (for the SPO256-AL2)
1	SW1,SW2	DPDT on/off power switch
1	14	* 3.12-MHz ceramic resonator (blue, CSA3.12MS2)
4 Resistors:		
2	5,8	33-kΩ (orange, orange, orange)
1	9	10-kΩ (brown, black, orange) or 100-kΩ (brown, black, yellow)
1	17	10-Ω (brown, black, black) or 11-Ω (brown, brown, black)

QUANTITY	PC BOARD IDENTIFICATION FOR THE COMPLETE KIT	DESCRIPTION
12 Capacitors:		
3	10,12,18	0.1 μF
1	11	100-μF electrolytic (audio filter)
1	6	10-μF electrolytic (power filter)
1	3	1-μF electrolytic
1	16	10-μF electrolytic (audio filter)
1	19	100-μF electrolytic (power filter)
2	4,7	0.025 μF
2	13,15	100 pF
Additional parts required (not included)		
1		7.5 to 9-V dc power supply (250–300mA). Can be obtained at Radio Shack, Catalog #2731455 or Catalog #603053.
1		Speaker—any 4- or 8-Ω speaker may be used; the choice is yours.

* The only parts included in the starter kits.

have them around your lab, any electronics store (e.g., Radio Shack) will surely have them. The starter kit is the least expensive type of kit; however, it requires the most amount of time to build the circuit. If you are not familiar with wiring up breadboards, you may want to try the complete kit or complete module. The starter kits were designed for the more experienced hobbyist who wants to save a little extra money in exchange for a portion of his or her time.

The complete kits consist of all the parts listed in Table 6-1, including printed circuit (PC) boards, connectors, and cabinet.

The speaker and the power supply are the only parts that have to be supplied. This kit is somewhat more expensive, but offers the ease of building the circuit on a PC board specifically designed for your particular computer. No wiring experience is necessary. All you need to know is how to use a soldering iron. This circuit is designed for the beginner who wants to learn how to build his or her own circuit and at the same time save a little money.

If after reading the instructions you are not interested in building a synthesizer yourself, you can purchase a complete module specifically designed for your computer. The complete modules are fully assembled and tested, and plug directly into the various computers, enabling them to speak in a matter of minutes!

NOTE: The parts that form the starter kits, complete kits, and complete modules can be purchased from a number of distributors listed in the Parts Supplier Listing, Appendix B. With some distributors you may have to purchase all the parts separately, with others you may be able to purchase the parts in kit or module form. (Refer to this listing and note the parts that each company supplies before setting out to purchase your speech synthesis kits.)

All circuits, kit or module form, conform to certain requirements; they are discussed below.

PORT LOCATIONS AND POWER SPECIFICATIONS

When plugged into the TRS-80 Printer Port Interface Model I, this board utilizes port 14312. When plugged into Models III and IV, it utilizes port 248. This board does not provide internal power, so the board *must* be powered externally.

PARTS LISTING

The list in Table 6-1 describes all parts required to build a speech synthesizer for your computer. Recall that the parts denoted by an asterisk combine to form the starter kits. All additional parts

must be supplied and wired up on an experimenter's breadboard. The complete kits include all the parts in the list with the exception of a speaker and power supply. The PC board identification (included for complete kits only) describes where the parts should be placed on the PC board (see the section on Complete Kit Assembly Instructions).

STARTER KIT ASSEMBLY INSTRUCTIONS

When building the starter kit, refer to the schematic diagram in Fig. 6-1.

> NOTE: After completing your starter kit, see the section on Complete Module and Operating Instructions.

STEP 1 In addition to the starter kit, obtain all of the components listed in the TRS-80 Printer Port Interface Models I, III, and IV Parts List, Table 6-1.

STEP 2 Insert the sockets into the breadboard and make a note of where pin 1 is to be located. (Pin 1 should be in the upper left-hand corner.)

STEP 3 Connect the ribbon cable to the 34-pin female connector. The connector states what the pin numbers are.

STEP 4 Solder or wirewrap the following connections:

FROM THE 34-PIN CONNECTOR:	*TO:*
GND pin 2	Common ground of the synthesizer board
D0 pin 3	Pin 18 of the SPO256-AL2
D1 pin 5	Pin 17 of the SPO256-AL2
D2 pin 7	Pin 16 of the SPO256-AL2
D3 pin 9	Pin 15 of the SPO256-AL2
D4 pin 11	Pin 14 of the SPO256-AL2
D5 pin 13	Pin 13 of the SPO256-AL2
D7 pin 17	Pin 20 of the SPO256-AL2
BUSY pin 21	Pin 9 of the SPO256-AL2

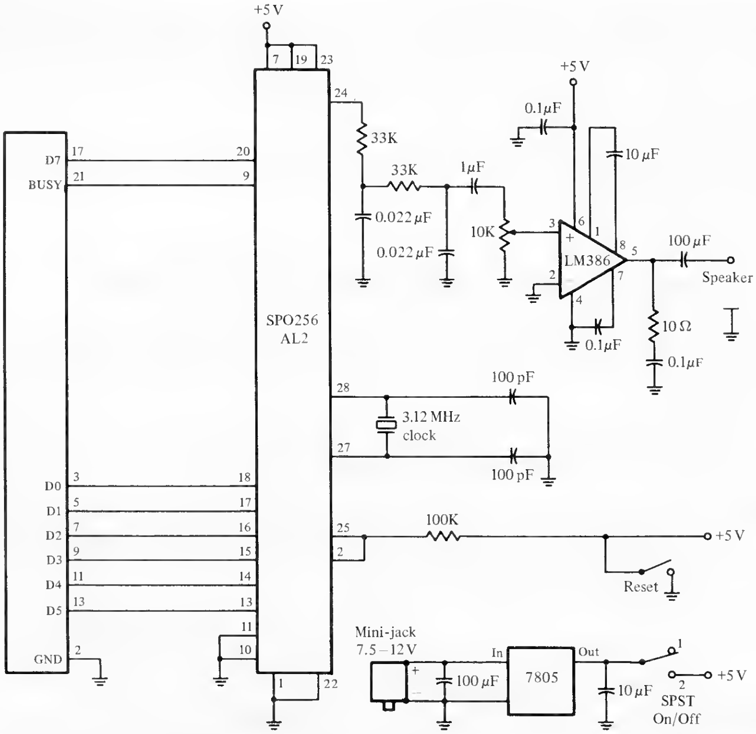


Fig. 6-1 TRS-80 Models I, III, and IV printer port interface speech circuit schematic.

STEP 5 The Power and Ground Connections

FROM SIDE 2 OF THE SPST ON/OFF SWITCH	*TO:*
	Pins 7, 19, and 23 of the SPO256-AL2
FROM PIN 2 OF THE 34-PIN CONNECTOR	*TO:*
	Pins 1 and 22 of the SPO256-AL2
	Pin 11 and 10 of the SPO256-AL2

STEP 6 The Reset Circuit

a. Connect a 100-kΩ resistor between +5 V (output of the 7805 or LM340) and pins 2 and 25 of the SPO256-AL2.
b. Connect the reset switch between ground (pin 2 of the 34-pin connector) and pins 2 and 25 of the SPO256-AL2.

STEP 7 The Regulator Circuit

a. Connect the positive side of the power jack to the 9-V input side of the 7805 (or LM340).
b. Connect the negative side of the power jack to the negative terminal of the 7805 (or LM340).
c. Connect the negative terminal of the 7805 (or LM340) to pin 2 of the 34-pin connector.
d. Connect the 5-V output of the 7805 (or LM340) to side 1 of the SPST on/off switch.
e. Connect the positive side of a 100-μF electrolytic capacitor between the 9-V input of the 7805 (or LM340) and the negative side to ground (pin 2 of the 34-pin connector).
f. Connect the positive side of a 10-μF electrolytic capacitor between the 5-V output of the 7805 (or LM340) and the negative side to ground (pin 2 of the 34-pin connector).

STEP 8 The Oscillator Circuit

a. Connect the 3.12-MHz ceramic resonator between pins 27 and 28 of the SPO256-AL2.
b. Connect a 100-pF capacitor between pin 27 of the SPO256-AL2 and ground (pin 2 of the 34-pin connector).
c. Connect a 100-pF capacitor between pin 28 of the SPO256-AL2 and ground (pin 2 of the 34-pin connector).

STEP 9 The Audio Filter Circuit

a. Connect side 1 of a 33-kΩ resistor to pin 24 of the SPO256-AL2.
b. Connect side 2 of the first 33-kΩ resistor to side 1 of the second 33-kΩ resistor and side 1 of the first

0.022-μF capacitor. Connect side 2 of the first 0.022-μF capacitor to ground.

c. Connect side 2 of the second 33-kΩ resistor to the positive side of the 1-μF electrolytic capacitor and to side 1 of the second 0.022-μF capacitor. Connect side 2 of the 0.022-μF capacitor to ground.
d. Connect the negative side of the 1-μF electrolytic capacitor to side 1 of the 10K potentiometer.
e. Connect side 2 (middle terminal) of the 10K potentiometer to pin 3 of the LM386.
f. Connect side 3 of the 10K potentiometer to ground.
g. Connect pins 2 and 4 of the LM386 to ground.
h. Connect a 0.1-μF capacitor between pins 4 and 7 of the LM386.
i. Connect the positive side of the 10-μF electrolytic capacitor to pin 1 of the LM386 and the negative side to pin 8 of the LM386.
j. Connect pin 6 of the LM386 to +5 V.
k. Connect a 0.1-μF capacitor between pin 6 of the LM386 and ground.
l. Connect pin 5 of the LM386 to side 1 of a 10-Ω resistor. Connect side 2 of the 10-Ω resistor to side 1 of a 0.1-μF capacitor. Connect side 2 of the 0.1-μF capacitor to ground.
m. Connect the positive side of the 100-μF electrolytic capacitor to pin 5 of the LM386. Connect the negative side of the 100-μF capacitor to side 1 of the speaker jack. Connect side 2 of the speaker jack to ground.

STEP 10 Insertion of the Integrated Circuits

NOTE: When inserting the IC's, be sure that pin 1 of the IC lines up with pin 1 of the socket that was previously labeled.

a. Insert the SPO256-AL2 into the 28-pin socket.
b. Insert the LM386 into the 8-pin socket.

STEP 11 Proceed to the section on the Complete Module and Operating Instructions.

COMPLETE KIT ASSEMBLY INSTRUCTIONS

When constructing the complete kits refer to Fig. 6-2. It illustrates the PC board and component placements. The **circuit side** of the board is the side where all the solder lines appear. (It is this side where all the components are to be soldered.) The other side (or flip side) is known as the **component side.** All the components are to be mounted or inserted into the appropriate holes on this side of the board, except where noted differently in the instructions.

The parts for each speech synthesizer kit are shown in Table 6-1. The PC board identification of the components refers to the place on the PC board where the components should be inserted.

For example, in Table 6-1 the 33-kΩ resistors are labeled 5 and 8. If you look at Fig. 6-2, you will notice that there are two places labeled 5 and 8. These are the spots where the resistors are to be inserted.

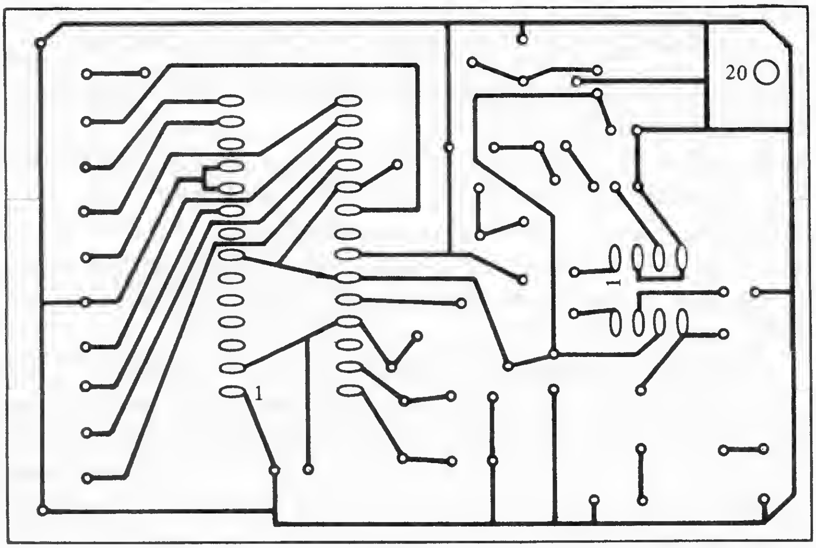
NOTE: After completing your synthesizer kit see the section on the Complete Module and Operating Instructions.

STEP 1 The PC Board Construction
Insert components 1 through 20 in the appropriate PC board locations denoted on the component side of the board. After insertion, solder the components onto the circuit side of the PC board.

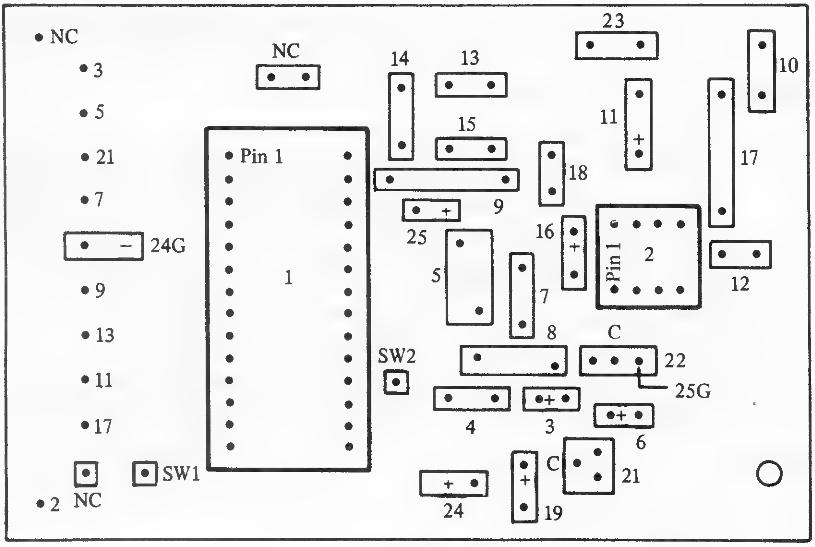
NOTE: NC, SW1, and SW2 are holes that are *not* to be used. Numbers 3, 6, 11, 16, and 19—the 1-μF, 10-μF, and 100-μF capacitors are all electrolytic. The polarities are denoted on the capacitors. The positive side is inserted in the holes on the component diagram labeled with a plus sign.

STEP 2 The Regulator

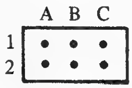
PC BOARD IDENTIFICATION NUMBER	*DESCRIPTION*
20	Ground plane
21	7805 (or LM340) 5-V regulator



Circuit side



Component side



DPDT switch

Fig. 6-2 TRS-80 printer port PC board speech circuit.

This part must be inserted into the hole labeled 21 and mounted onto the circuit side of the PC board. The metal back of the regulator should be able to rest on the PC board ground plane labeled 20 (see circuit side diagram). Secure the regulator with the retaining bolt and nut.

NOTE: The connections are still soldered on the circuit side of the board. The "C" indicates the center connection.

STEP 3 The Connector Connections

The pin numbers labeled on the left-hand side of the component diagram correspond to the pin numbers of the TRS-

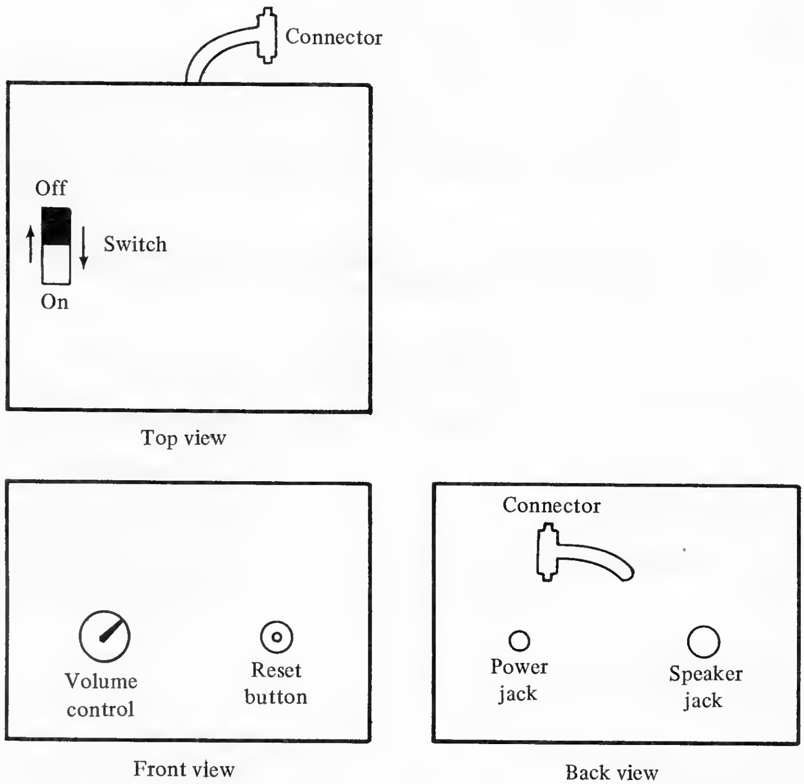


Fig. 6-3 TRS-80 printer port component placement.

80 printer port bus as specified in the TRS-80 technical reference manual. Connect the ribbon cable from the connector to the PC board in this manner.

NOTE: Do not connect hole 24G to the ribbon cable, refer to Step 5 for this connection.

STEP 4 Insertion of the Integrated Circuits

a. After soldering the 28-pin socket, insert the SPO256-AL2. Pin 1 of the SPO256-AL2 is denoted by the dot in the corner of the chip and is labeled on the diagram.
b. After soldering the 8-pin socket, insert the LM386. Pin 1 of the LM386 is denoted by the dot in the corner of the chip and is labeled on the diagram.

STEP 5 Case Assembly Mountings

The following components must be mounted in the case assembly as shown in Fig. 6-3. After mounting these components into the box assembly, connect them to the following locations specified on the component diagram in Fig. 6-2.

NOTE: "C" denoted on the component diagram represents the center connection for the potentiometer.

COMPONENT TO BE MOUNTED	*CONNECT TO PC BOARD IDENTIFICATION NUMBER*	*DESCRIPTION*
10K potentiometer	22	Volume control
$\frac{1}{4}''$ Speaker jack	23	Speaker connections
"Mini" power jack tip	Side 1B of the DPDT switch	Positive 9-V input power jack connection
"Mini" power jack ground	24G	Negative 9-V input power jack connection
Side 1C of the DPDT switch	24	Positive 9-V input to PC board

COMPONENT TO BE MOUNTED	*CONNECT TO PC BOARD IDENTIFICATION NUMBER*	*DESCRIPTION*
Reset switch (side 1)	25	Reset switch connection
Reset switch (side 2)	25G	Reset switch connection

STEP 6 Case Assembly

a. Secure the PC board into the box assembly with silicone sealer.

b. Install the bottom of the case onto the top of the case with the 4 holding screws.

COMPLETE MODULE AND OPERATING INSTRUCTIONS

Whether you have built the starter kit or complete kit, or purchased the complete module, you are now ready to operate your speech synthesizer (see Fig. 6-3).

Reset Button—Depressing this button readies the synthesizer for operation. It will also cause the board to stop talking.

Volume Control—Turning this control to the right or left will increase or decrease the volume of the synthesizer.

Speaker Jack—This connection will drive any 4- or 8-Ω speaker or act as an auxiliary input to any receiver.

Power Jack—Power connection to be used at all times (see the section on Port Locations and Power Specifications at the beginning of this chapter). The power supply required is a 9-V, 300-mA supply, mini-jack (the tip is positive).

34-Pin Dual Edge Connector—This connector is the interface between the synthesizer and the TRS-80 Printer Port.

On/Off Switch—turns power supply on or off.

Operating Procedure

STEP 1 Switch the on/off switch to the off position.
STEP 2 Plug the speaker into the speaker jack.
STEP 3 Plug the external power supply into the power jack of the synthesizer (the tip is positive).

NOTE: Do not plug into wall outlet yet.

CAUTION: The computer must be off.

STEP 4 Plug the synthesizer into the 34-pin bus of the TRS-80 printer port.
STEP 5 Plug in the 9-V power adaptor to the synthesizer into the wall outlet.
STEP 6 Turn your computer on.
STEP 7 Turn the synthesizer switch to the on position.
STEP 8 Depress the reset button.

NOTE: A faint "click" should be heard in the speaker when depressing this button. If this sound is not heard, turn up the volume and try it again. If this condition still exists, proceed to the following section.

TESTING YOUR CIRCUIT

After carefully wiring up your speech synthesizer board, the following commands can be used to test your circuit before loading in the respective programs. Before power is applied, visually inspect your hardware to ensure that the proper connections have been made and all the grounds are secure. On power up, a hardware reset is required—simply close the switch momentarily. A click or pop should be heard in the speaker. If this occurs, proceed with the instructions in this section. If this condition does not occur, refer to the section on Debugging Your Hardware at the end of this chapter.

TRS-80 Printer Port Models III and IV

When testing this circuit, you must first initialize $\overline{\text{ALD}}$. To do so, enter the statement OUT 248,128. Then enter the statement OUT 248,133. This will output address 5 (the OY allophone) on the data bus of the speech synthesizer maintaining a logic 1 on $\overline{\text{ALD}}$. The statement OUT 248,5 will maintain address 5 on the data lines and bring $\overline{\text{ALD}}$ low, causing the synthesizer to talk. To silence the speech synthesizer, the following commands are necessary:

OUT 248,128	This brings the $\overline{\text{ALD}}$ line high again.
OUT 248,0	This outputs address 000 which is a pause. It also brings $\overline{\text{ALD}}$ low.
OUT 248,128	This brings the $\overline{\text{ALD}}$ line high again to its normal state of operation.

You are now ready to load in your Exclusive Program (in the usual manner as specified in your computer manual) and create your own allophone phrases.

TRS-80 Printer Port Model I

When testing this circuit, you must first initialize $\overline{\text{ALD}}$. To do so, enter the statement POKE 14312,128. Then enter the statement POKE 14312,133. This will output address 5 (the OY allophone) on the data bus of the speech synthesizer maintaining a logic 1 on $\overline{\text{ALD}}$. The statement POKE 14312,5 will maintain address 5 on the data lines and bring $\overline{\text{ALD}}$ low causing the synthesizer to talk. To silence the speech synthesizer, the following commands are necessary:

POKE 14312,128	This brings the $\overline{\text{ALD}}$ line high again.
POKE 14312,0	This outputs address 000 which is a pause. It also brings $\overline{\text{ALD}}$ low.

POKE 14312,128 This brings the $\overline{\text{ALD}}$ line high again to its normal state of operation.

You are now ready to load in your Exclusive Program (in the usual manner as specified in your computer manual) and create your own allophone phrases.

SOFTWARE

The Exclusive Phrase Finder Programs that follow allow you to build words and phrases from their constituent allophones. The phrase can be edited by moving a pointer left or right to the desired position (see Table 6-2). Inserting, deleting, or replacing allophones can then be accomplished quite easily. When the phrase is prepared to your satisfaction, a simple ENTER, NEW LINE, or RETURN will signal the synthesizer to talk.

Table 6-2 ALLOPHONE EDITING COMMANDS

AVAILABLE COMMANDS	OPERATION
"phoneme strings"	Causes named allophone to be added to the phrase at the current position of the pointer, by either replacing the existing allophone or inserting one before it (see the section on Inserting an Allophone).
"L"	Moves the position pointer left one allophone.
"R"	Moves the position pointer right one allophone.
"D"	Deletes allophone at the current position pointer.
NEW LINE, ENTER, or RETURN	Causes the system to output to the hardware the commands necessary to pronounce the phrase.
"E"	Exits the program.

AVAILABLE COMMANDS	OPERATION
"XL"	Moves the position pointer "X" number of allophones to the left.
"XR"	Moves the position pointer "X" number of allophones to the right.
"I"	Turns on the "insert mode." The next allophone entered will be inserted into the phrase at the current position of the pointer. Additional allophones will be inserted until "I" is entered again. The second "I" command will turn the insert mode off. When the insert mode is off, an entered allophone will replace the one at the current position. This is the default at system start-up.

EXCLUSIVE PHRASE FINDER PROGRAM DESCRIPTION

At system start-up, a brief message will be spoken and the following commands are performed.

> NOTE: The message will be spoken each time the program is RUN. To delete this message from the system, the command "Delete line 75 from the program" must be issued.

The screen is cleared and all variables are initialized. The allophone symbol array is initialized with the 64 two- or three-character symbols that represent each allophone. These are strings the user will enter in order to add an allophone to the phrase (see Tables 3-1 and 9-1). The position pointer is at position one, and the user is prompted for input with a "≥." At this prompt, an allophone or any of the commands in Table 6-2 may be entered. An invalid allophone will be flagged as an error, as will attempting to move the position pointer to the left or right of the boundaries of the

phrase. After each command, the updated phrase is displayed with the current position indicated by "⩾."

For example, at system start-up, the screen will look like this:

COMMAND	*SCREEN*	*COMMENTS*
RUN	⩾ ?	Pointer at position one.
HH1	⩾ ?HH1	Desired allophone.
ENTER	HH1⩾ ?	The first allophone has been entered; the pointer is at position two; the system is ready for the next input.
EH1	HH1⩾ ?EH1	Next allophone.
ENTER	HH1⩾ ?EH1 ***INVALID ENTRY***	User entered invalid data.
EH	HH1⩾ ?EH	
ENTER	HH1 EH⩾	The second allophone has been entered; the pointer is at position three; the system is ready for the next input.

Editing Features

Upon entering a string of allophones and noticing that a few corrections are in order, the following edit commands are useful.

First, we must position the pointer at the location where an

editing command is to be performed. Let's take the word "hello" for example. The screen should now look like this:

HH1 EH LL UW1>

Realizing that the UW1 allophone is incorrect, we would like to REPLACE it with "OW." To do so, the following commands are required:

Replacing an Allophone

COMMAND	*SCREEN*	*COMMENTS*
"L"	HH1 EH LL UW1> L	We have to move the pointer left one space to replace UW1.
"ENTER"	HH1 EH LL> UW1	The system is now ready to replace the allophone UW1.
"OW"	HH1 EH LL> UW1 OW	The desired replacement allophone is typed.
"ENTER"	HH1 EH LL OW>	After pressing ENTER, the allophone has been replaced.

Moving the position pointer right works in the same manner as moving to the left. The only exception is that we use an "R" instead of an "L." These commands move the pointer one space at the time.

In replace mode (the default at program start-up), the new allophone will replace the allophone at the current position in the phrase.

NOTE: If you attempt to move the position pointer left or right beyond its boundaries, the message ***INVALID ENTRY*** will appear.

COMMAND	*SCREEN*	*COMMENTS*
"R" ENTER	HH1 EH LL OW> R ***INVALID ENTRY***	To clear the invalid entry, press ENTER or type in a valid command.
"L" ENTER	>HH1 EH LL OW L ***INVALID ENTRY***	

The XL and XR commands work in the same manner as the L and R commands. The only difference is that these commands move the position pointer "X" number of allophones (or spaces) to the left or right. Once again, exceeding the boundaries will prompt an ***INVALID ENTRY*** message. For example:

COMMAND	*SCREEN*	*COMMENTS*
"2L"	HH1 EH LL OW1>	
ENTER	HH1 EH> LL OW	The position pointer has moved 2 spaces to the left.

Deleting an Allophone. We'll use the same examples as above. Remember, we must first position the pointer to the specified allophone to be deleted. Once this is accomplished, the following commands are required.

COMMAND	*SCREEN*	*COMMENTS*
"L" ENTER (3 times)	HH1 EH EH LL OW>	We need to delete an EH here, so first we must move the pointer 3 spaces to the left.
"D"	HH1 EH> EH LL OW D	The pointer is now positioned to the allophone to be deleted.

COMMAND	*SCREEN*	*COMMENTS*
"ENTER"	HH1 EH> LL OW	The EH allophone has been deleted.

NOTE: The delete command "D" deletes one allophone at a time.

Inserting an Allophone. After creating your allophone phrases, and realizing that a few pauses (or other allophones) need to be inserted, the following command sequence must be performed. Let's use the word "chatter" as an example.

COMMAND	*SCREEN*	*COMMENTS*
"L" ENTER (3 times)	CH AE TT2 ER1 PA3> ?	We must position the pointer at the location where the inserted allophone will go.
"I"	CH AE> TT2 ER1 PA3 ?I	We are ready to turn on the "insert mode."
"ENTER"	CH AE> TT2 ER1 PA3 ?	Insert mode has been turned on.
"PA3"	CH AE> TT2 ER1 PA3 ? PA3	The desired allophone to be inserted is typed.
"ENTER"	CH AE PA3> TT2 ER2 PA3 ?	The allophone has been inserted. Additional allophones may be entered at this time if required.
"I"	CH AE PA3> TT2 ER2 PA3 ?I	We are ready to turn off the insert mode.

COMMAND	*SCREEN*	*COMMENTS*
"ENTER"	CH AE PA3> TT2 ER2 PA3 ?	Insert mode has been turned off.

MAKING YOUR COMPUTER TALK

A simple "ENTER," "RETURN," or "NEW LINE" is all that's required to make the system talk. If a new allophone string is desired, you must first EXIT (E) the program and then RUN it again. This will clear all the previous allophone codes stored. Unfortunately at the time of writing this book, a "SAVE" routine was not supplied with the Exclusive Phrase Finder Program. You may want to try writing your own subroutine to save the allophone strings you've created with the program (in separate files). An alternative method is to save the Exclusive Phrase Finder Program on cassette tape or disk, with the phrase or word you've just created. When reloading the program, the following command is necessary for it to speak the prestored phrase.

```
GOTO    300
```

Upon typing this command, the computer will speak the prestored allophone string and the allophone codes will be printed on the screen. The system is now ready to edit the existing string. To enter a new string, RUN the program.

EXCLUSIVE PHRASE FINDER PROGRAM LISTINGS

TRS-80 Printer Port Interface Models III and IV

NOTE: When typing in the allophones, it is essential to type blank spaces in place of all **b**'s in the program below.

```
  0    CLS
  2    PRINT "***PHONEME CONCATENATION SYSTEM***"
  5    CLEAR 1000
  7    OUT 248, 128
 10    DIM NA$(64) : REM PHONEME SYMBOL ARRAY
 12    DIM PH$(64) : REM CURRENT PHONEME STRING
 14    DIM PH(64)  : REM CURRENT PHONEME STRING ADDRESSES
 16    S$ = ""     : REM DISPLAY STRING, WITH CURSOR POSITION
 18    I$ = ""     : REM INPUT STRING FROM USER
 25    P=1         : REM CURRENT POSITION POINTER
 30    L=0         : REM LENGTH OF PHONEME STRING
 35    IN=0        : REM "INSERT MODE" FLAG (0=OFF, 1=ON)
 50    DATA PA1,PA2,PA3,PA4,PA5,OY,AY,EH,KK3,PP,JH,
       NN1,IH,TT2,RR1,AX,MM,TT1,DH1,IY,EY,DD1,UW1
 51    DATA AO,AA,YY2,AE,HH1,BB1,TH,UH,UW2,AW,DD2,
       GG3,VV,GG1,SH,ZH,RR2,FF,KK2,KK1,ZZ,NG,LL
 52    DATA WW,XR,WH,YY1,CH,ER1,ER2,OW,DH2,SS,NN2,
       HH2,OR,AR,YR,GG2,EL,BB2
 55    FOR X=1 TO 64
 60    READ NA$(X)
 70    NEXT X
 75    GOTO 450    : REM PROGRAM START UP TALK
 98    REM ***MAIN INPUT/DECODE LOOP***
100    GOSUB 400   : REM DISPLAY CURRENT STRING
101    IF IN=1 THEN PRINT "**Insert Mode**"
103    I$="" : J=1
105    INPUT I$
106    IF I$=""  GOTO 300 : REM TALK
108    IF I$="E" THEN END : REM EXIT
109    IF I$="L" GOTO 115 : REM MOVE LEFT
110    IF I$="R" GOTO 121 : REM MOVE RIGHT
111    IF I$="D" GOTO 131 : REM DELETE PHONEME
112    IF I$="I" GOTO 140 : REM INSERT PHONEME
113    IF ASC(I$)<58 THEN 1000
114    GOTO 150
115    REM ***MOVE LEFT***
116    IF P-J=0 GOTO 165
117    P=P-J
119    GOTO 100
121    REM ***MOVE RIGHT***
123    IF P+J>L+1 GOTO 165
125    IF P+J>64 GOTO 165
126    P=P+J
129    GOTO 100
131    REM ***DELETE PHONEME***
132    IF P>L GOTO 100
133    FOR I=P TO L
134    PH$(I)=PH$(I+1)
135    PH(I)=PH(I+1)
136    NEXT I
137    L=L-1
139    GOTO 100
140    REM ***TURN ON "INSERT MODE"***
145    IN=ABS(IN-1)
147    GOTO 100
150    REM ***TEST FOR VALID PHONEME ENTRY***
152    FOR X=1 TO 64
155    IF I$=NA$(X) GOTO 190
```

```
160     NEXT X
165     REM ***USER ENTERED INVALID DATA***
168     PRINT "***INVALID ENTRY***"
170     GOTO 103 : REM RETURN TO MAINLINE,
        DON'T REFRESH SCREEN
190     REM ***MODIFY STRING***
195     X=X-1
200     IF P>64 GOTO 165
201     IF IN=0 GOTO 210 : REM TEST FOR "INSERT MODE"
        BEING ON
202     IF L>63 GOTO 165 : REM ERROR-NO ROOM TO INSERT
203     FOR J=L+1 TO P+1 STEP-1
204     PH$(J) = PH$(J-1)
205     PH(J) = PH(J-1)
206     NEXT J
208     L=L+1
210     PH$(P)=I$
215     PH(P)=X
216     IF L<P THEN L=P
217     P=P+1
225     GOTO 100
300     REM ***TALK*** FOR THE MODEL III AND IV
305     IF L<1 GOTO 100
310     OUT 248, 128
320     FOR J=1 TO L
325     OUT 248, 128 + PH(J)
330     OUT 248, PH(J)
335     OUT 248, 128
350     IF PEEK(14312)  = 128 GOTO 350
355     NEXT J
360     OUT 248, 0
370     OUT 248, 128
385     IF FIRST = 1 THEN GOTO 490
390     GOTO 100
400     REM ***BUILD AND DISPLAY PHONEME STRING***
403     S$=""
405     IF P=1 THEN S$="→"
408     FOR X = 1 TO L
410     S$ = S$ + PH$(X) + "b"
415     IF X = P-1 THEN S$ = S$ + "→"
420     NEXT X
425     CLS
430     PRINT a 256,S$
435     RETURN
440     END
450     DATA 27,7,45,53,4,4,4,6,2,42,26,11,2,36,12,
             35,2,29,19,2,9,32,51,2,15,15,35,2
451     DATA 55,0,9,19,0,50,2,13,31,2,25,58,2,42,
             15,16,0,9,49,22,0,13,51,4
453     RL=52
455     FOR X = 1 TO RL
457     READ PH(X)
459     NEXT X
460     L = RL
470     FIRST = 1
480     GOTO 300
490     L = 1: FIRST = 0
500     GOTO 100
```

```
1000      LET J=VAL(I$)
1010      IF RIGHT$(I$,1) = "R" THEN 121
1020      IF RIGHT$(I$,1) = "L" THEN 115
1030      GOTO 165
```

TRS-80 Printer Port Interface Model I

NOTE: When typing in the allophones, it is essential to type a blank space in place of all **b**'s in the program below.

Follow the program listing for the TRS-80 Printer Port Interface Models III and IV except that lines 300–400 should be replaced with the following:

```
300      REM ***TALK*** FOR THE MODEL I
305      IF L<1 GOTO 100
310      POKE 14312, 128
320      FOR J=1 TO L
325      POKE 14312, 128 + PH(J)
330      POKE 14312, PH(J)
335      POKE 14312, 128
350      IF PEEK(14312)  = 128 GOTO 350
355      NEXT J
360      POKE 14312, 0
370      POKE 14312, 128
385      IF FIRST = 1 THEN GOTO 490
390      GOTO 100
```

SAMPLE PROGRAM

The following program describes how to add "N" phrases to your existing program.

Data Statements

These statements must appear in the program before the lines that enable the synthesizer to speak.

NOTE: The decimal codes for each particular allophone (as shown in Table 9-1) must be inserted in the places denoted by an asterisk; 128 must be added to the *last* allophone code in each data statement.

```
10    DEFINT A,N,X
20    DIM A(10,255): REM 10 represents the number of phrases
      to be spoken.  255 represents the number of allophones in
      the longest phrase.
100   FOR N=1 TO 10: REM 10 equals the number of phrases.
110   X=1
120   READ A
130   A(N,X)=A
140   X=X+1
150   IF (A and 128)=0 THEN 120
160   NEXT N
200   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      first phrase.
210   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      second phrase.
220   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      third phrase.
230   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      fourth phrase.
240   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      fifth phrase.
250   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      sixth phrase.
260   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      seventh phrase.
270   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      eighth phrase.
280   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      ninth phrase.
290   DATA *,*,*, . . . ,*: REM decimal codes of allophones for
      tenth phrase.
```

Subroutine for the TRS-80 Printer Port Interface Model I

```
1000  POKE 14312,128
1010  X=1
1020  IF (PEEK(14312) AND 128)=128 GOTO 1020
```

```
1030    POKE 14312,128 OR A(N,X)
1040    POKE 14312,127 AND A(N,X)
1050    POKE 14312,128
1070    IF (A(N,X) AND 128)=0 THEN X=X+1: GOTO 1020
1080    IF (PEEK(14312) AND 128)=128 THEN 1080
1090    POKE 14312,0
1100    POKE 14312,128
1110    RETURN
```

Subroutine for the TRS-80 Printer Port Interface Models III and IV

```
1000    OUT 248,128
1010    X=1
1020    IF (PEEK(14312) AND 128)=128 GOTO 1020
1030    OUT 248,128 OR A(N,X)
1040    OUT 248,127 AND A(N,X)
1050    OUT 248,128
1070    IF (A(N,X) AND 128)=0 THEN X=X+1: GOTO 1020
1080    IF (PEEK(14312) AND 128)=128 THEN 1080
1090    OUT 248,0
1100    OUT 248,128
1110    RETURN
```

First Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the first phrase.

```
1000    LET N=1
1010    GOSUB 10000
```

Second Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the second phrase.

```
2000    LET N=2
2010    GOSUB 10000
```

Third Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the third phrase.

```
3000    LET N=3
3010    GOSUB 10000
```

Nth Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the Nth phrase.

```
N000    LET N=N
N010    GOSUB 10000
```

LET'S GET TECHNICAL . . .

DRIVER SUBROUTINES

This next section describes the driver, or talk subroutines, for the TRS-80 Printer Port Interface Models I, III, and IV. These driver subroutines can be used directly in your own application programs, or studied as examples.

TRS-80 Printer Port Interface Models III and IV

```
305    IF L<1 GOTO 100
310    OUT 248,128
320    FOR J=1 TO L
325    OUT 248,128+PH(J)
330    OUT 248, PH(J)
335    OUT 248,128
350    IF PEEK(14312)>=128 GOTO 350
355    NEXT J
360    OUT 248,0
370    OUT 248,128
```

```
385     IF FIRST=1 THEN GOTO 490
390     GOTO 100
```

Explanation of the Program

Line 305 IF L<1 GOTO 100

L<1 is an illegal length of string. If L<1, the system will require a different input.

Line 310 OUT 248,128

This statement initializes the address load to a logic 1 (see Address Load in the hardware section of Chapter Nine).

Line 320 FOR J=1 TO L

L represents the length of the phoneme string to be spoken. J represents the loop counter.

Line 325 OUT 248,128+PH(J)

This statement outputs the allophone address (see Table 9-1) onto the address lines of the speech synthesizer. Bit 7 used for the address load; pulse is still high.

Line 330 OUT 248, PH(J)

The address load is strobed causing the speech synthesizer to talk.

Line 335 OUT 248,128

The address load is set back to a logic 1.

Line 350 IF PEEK(14312)>=128 GOTO 350

The busy line (location 14312) is connected to the load request (see Load Request in the hardware section of Chapter Nine). When this line is high, the speech synthesizer is still talking and cannot accept the next address.

Line 355 NEXT J

When the load request goes low, the next address may be loaded.

Line 360 OUT 248,0

Line 370 OUT 248,128

These statements output a pause on the address lines to end the allophone phrase.

Line 385 IF FIRST=1 THEN GOTO 490

Due to the message spoken at program start-up, L was set equal to 52. This statement sets L back to 1.

Line 390 GOTO 100

The system is ready for the next input.

TRS-80 Printer Port Interface Model I

```
305     IF L<1 GOTO 100
310     POKE 14312,128
320     FOR J=1 TO L
325     POKE 14312,128+PH(J)
330     POKE 14312, PH(J)
335     POKE 14312,128
350     IF PEEK(14312)>=128 GOTO 350
355     NEXT J
360     POKE 14312,0
370     POKE 14312,128
385     IF FIRST=1 THEN GOTO 490
390     GOTO 100
```

Explanation of the Program. Model I follows the explanation of the program for Models III and IV except all OUT 248 statements should be changed to POKE 14312 statements.

DEBUGGING YOUR CIRCUIT

The following steps are simple test procedures to follow when debugging your speech synthesis circuit. Please note that additional test equipment may be required (e.g., oscilloscope, voltmeter, logic

probe). In addition to the test equipment, you should be experienced with debugging electrical circuits.

Procedures to Ensure That Your Connections Are Correct

1. Check all the power and ground connections. Pins 7, 19, and 23 of the SPO256-AL2 are +5 V. Pins 2 and 22 of the SPO256-AL2 are negatively grounded.
2. Disconnect the audio filter circuit from pin 24. Turn up the volume on the LM386 to maximum. Send another audio signal through the audio filter circuit. The output of a transistor radio is sufficient. Tune into a station where speech is playing rather than music. If the signal is heard at the speaker, this circuit is OK (it may be distorted somewhat). When resetting the chip, pin 24 should go to a logic 0 level. When the device is not speaking, a 40-kHz square wave is present on this pin.
3. Pins 2 and 25 of the SPO256-AL2 should be a logic 1 (+5 V). Pressing the reset button should change it to a logic 0 (0 V) and a small click should be heard in the speaker. If this doesn't occur and Procedures 1 and 2 are OK, the problem has to be in the 100-kΩ resistor, the push button switch, or the SPO256-AL2.
4. The signal on pin 26 should be oscillating. If this does not occur, either the crystal is bad or the capacitors connected to ground are the wrong capacitance. An oscilloscope or logic probe is required for this test.
5. Disconnect all the address lines (pins 10, 11, 13–18) from your microcomputer and then ground the pins (except pin 15). Connect pin 15 to +5 V. Disconnect the $\overline{\text{ALD}}$ (pin 20) and connect a momentary switch between it and +5 v. Upon depressing the momentary switch, allophone KK3 should be heard. If Steps 1–4 are operating properly and this step fails, the SPO256-AL2 is malfunctioning.

6. If Steps 1–5 are operating, proceed with the following. For the TRS-80 Model I, enter POKE 14312,128 then POKE 14312,133 and POKE 14312,5. For the TRS-80 Models III and IV, enter OUT 248,128 then OUT 248,133 and OUT 248,5. Immediately on entering POKE 14312,5 or OUT 248,5 the state of the $\overline{\text{LRQ}}$ line (pin 9 of the SPO256-AL2) will change from a logic 0 to a logic 1. If this does not occur, carefully recheck all your wiring.

CHAPTER SEVEN

MAKING YOUR VIC 20 AND COMMODORE 64 TALK

The VIC 20 and Commodore 64 computers are the best of the low-end computers on the market. With the VIC 20 (under $100) aimed at the beginning computerist and the Commodore 64 (under $200) for the more-advanced computerist, the investment will be paid back ten-fold within a matter of weeks—both are ideal computers with which to incorporate speech.

This chapter consists of two sections: one describes the hardware interfaces, schematics, and parts required to make your Commodore computer talk. The other explains and provides the software required to drive the associated hardware. A detailed discussion of the program listing and its editing features is also included.

HARDWARE

This section explains all that's required to build a speech interface unit for your Commodore computer. It is divided into three parts: the *Starter Kit,* the *Complete Kit,* and the *Complete Module.*

The starter kits for each particular model contain the components denoted in Table 7-1 by an asterisk. (They usually include the speech synthesis chip, the ceramic resonator, and the TTL components.) The remaining components (experimenter's breadboard,

connector, ribbon cable, resistors, capacitors, etc.) have to be purchased separately. These parts are very popular and if you don't have them around your lab, any electronics store (e.g., Radio Shack) will surely have them. The starter kit is the least expensive type of kit; however, it requires the most amount of time to build the circuit. If you are not familiar with wiring up breadboards, you

Table 7-1 VIC 20 AND COMMODORE 64 PARTS LIST

QUANTITY	PC BOARD IDENTIFICATION FOR THE COMPLETE KIT	DESCRIPTION
1		PC board (3″ × 4″) or breadboard
1	25	Reset switch
1	24	Power jack (mini)
1	23	RCA phono speaker jack ($\frac{1}{4}$″)
1	22	10K potentiometer (volume control, 3-prong)
1		24-pin connector (female)
1	20,21	7805 C or LM340–5 5-V regulator
1		*SPO256-AL2 speech synthesizer
1		LM386–1 operational amplifier (This is required to drive a 4- or 8-Ω speaker.)
1	2	8-pin socket (for the LM386)
1	1	28-pin socket (for the SPO256-AL2)
1	SW1,SW2	DPDT internal on/off power switch
1	14	*3.12-MHz ceramic resonator (blue, CSA3.12MS2)
4 Resistors:		
2	5,8	33-kΩ (orange, orange, orange)
1	9	10-kΩ (brown, black, orange) or 100-kΩ (brown, black, yellow)

QUANTITY	PC BOARD IDENTIFICATION FOR THE COMPLETE KIT	DESCRIPTION
1	17	10-Ω (brown, black, black) or 11-Ω (brown, brown, black)
12 Capacitors:		
3	10,12,18	0.1 μF
1	11	100-μF electrolytic (audio filter)
1	6	10-μF electrolytic (power filter)
1	3	1-μF electrolytic
1	16	10-μF electrolytic (audio filter)
1	19	100-μF electrolytic (power filter)
2	4,7	0.025 μF
2	13,15	100 pF
Additional parts required (*not included*)		
1		7.5 to 9-V dc power supply (250–300mA) Optional—to be used only if external power is required. Can be obtained at Radio Shack, Catalog #2731455 or Catalog #603053.
1		Speaker—any 4- or 8-Ω speaker may be used; the choice is yours.

* The only parts included in the starter kits.

may want to try the complete kit or complete module. The starter kits were designed for the more experienced hobbyist who wants to save a little extra money in exchange for a portion of his or her time.

The complete kits consist of all the parts listed in Table 7-1, including printed circuit (PC) boards, connectors, and cabinet.

The speaker (and possibly the power supply) is the only part that has to be supplied. This kit is somewhat more expensive, but offers the ease of building the circuit on a PC board specifically designed for your particular computer. No wiring experience is necessary. All you need to know is how to use a soldering iron. This circuit is designed for the beginner who wants to learn how to build his or her own circuit and at the same time save a little money.

If after reading the instructions you are not interested in building a synthesizer yourself, you can purchase a complete module specifically designed for your computer. The complete modules are fully assembled and tested, and plug directly into the various computers, enabling them to speak in a matter of minutes!

> NOTE: The parts that form the starter kits, complete kits, and complete modules can be purchased from a number of distributors listed in the Parts Supplier Listing, Appendix B. With some distributors you may have to purchase all the parts separately, with others you may be able to purchase the parts in kit or module form. (Refer to this listing and note the parts that each company supplies before setting out to purchase your speech synthesis kits.)

All circuits, kit or module form, conform to certain requirements; they are discussed below.

PORT LOCATIONS AND POWER SPECIFICATIONS

This board plugs directly into the user port of the computer. The user port is the port on the left-hand side of the computer; the port opposite the game cartridge port. The VIC 20 always requires external power. The Commodore 64 supplies the bus with 250 mA and this is sufficient to drive the speech synthesizer. However, if another peripheral is used, internal power must be disconnected and external power applied.

PARTS LISTING

The list in Table 7-1 describes all parts required to build a speech synthesizer for your computer. Recall that the parts denoted by an asterisk combine to form the starter kits. All additional parts must be supplied and wired up on an experimenter's breadboard. The complete kits include all the parts in the list with the exception of a speaker and power supply. The PC board identification (included for complete kits only) describes where the parts should be placed on the PC board (see the section on Complete Kit Assembly Instructions).

STARTER KIT ASSEMBLY INSTRUCTIONS

When building the starter kit, refer to the schematic diagram in Fig. 7-1.

NOTE: After completing your starter kit, see the section on Complete Module and Operating Instructions.

STEP 1 In addition to the starter kit, obtain all of the components listed in the VIC 20 and Commodore 64 Parts List, Table 7-1.

STEP 2 Insert the sockets into the breadboard and make a note of where pin 1 is to be located. (Pin 1 should be in the upper left-hand corner.)

STEP 3 Connect the ribbon cable to the 24-pin female connector. The connector states what the pin numbers are.

STEP 4 Solder or wirewrap the following connections:

FROM THE 24-PIN CONNECTOR:	*TO:*
+5 V pin 2	Side 2A of the DPDT switch
GND pins A, N, and 12	Common ground of the synthesizer board
PB0 pin C	Pin 18 of the SPO256-AL2
PB1 pin D	Pin 17 of the SPO256-AL2
PB2 pin E	Pin 16 of the SPO256-AL2

FROM THE 24-PIN CONNECTOR:	*TO:*
PB3 pin F	Pin 15 of the SPO256-AL2
PB4 pin H	Pin 14 of the SPO256-AL2
PB5 pin J	Pin 13 of the SPO256-AL2
PB6 pin K	Pin 20 of the SPO256-AL2
PB7 pin L	Pin 9 of the SPO256-AL2

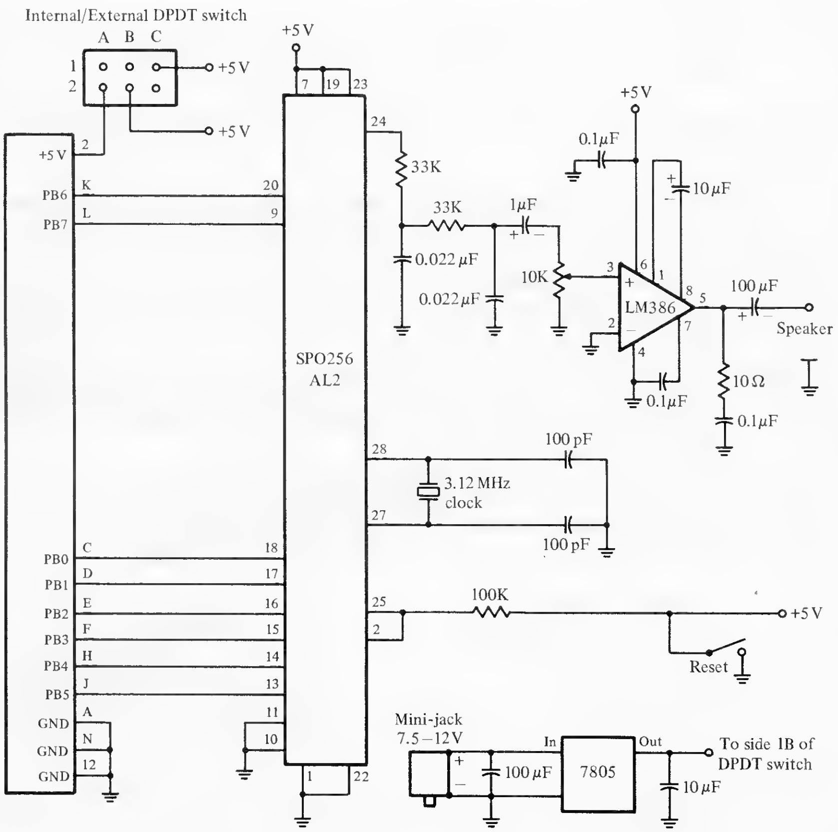


Fig. 7-1 VIC 20 and Commodore 64 speech circuit schematic.

STEP 5 The Power and Ground Connections

FROM SIDES 2B AND 1C OF THE DPDT SWITCH	*TO:*
	Pins 7, 19, and 23 of the SPO256-AL2
FROM PIN A, N, AND 12 OF THE 24-PIN CONNECTOR	*TO:*
	Pins 1 and 22 of the SPO256-AL2
	Pins 11 and 10 of the SPO256-AL2

STEP 6 The Reset Circuit

a. Connect a 100-kΩ resistor between +5 V (output of the 7805 or LM340) and pins 2 and 25 of the SPO256-AL2.

b. Connect the reset switch between ground (pin 12 of the 24-pin connector) and pins 2 and 25 of the SPO256-AL2.

STEP 7 The Regulator Circuit

a. Connect the positive side of the power jack to the 9-V input side of the 7805 (or LM340).

b. Connect the negative side of the power jack to the negative terminal of the 7805 (or LM340).

c. Connect the negative terminal of the 7805 (or LM340) to pin 12 of the 24-pin connector.

d. Connect the positive side of a 100-μF electrolytic capacitor between the 9-V input of the 7805 (or LM340) and the negative side to ground (pin 12 of the 24-pin connector).

e. Connect the positive side of a 10-μF electrolytic capacitor between the 5-V input of the 7805 (or LM340) and the negative side to ground (pin 12 of the 24-pin connector).

f. Connect the 5-V output of the 7805 (or LM340) to side 1B of the DPDT internal/external switch.

STEP 8 The Oscillator Circuit

a. Connect the 3.12-MHz ceramic resonator between pins 27 and 28 of the SPO256-AL2.
b. Connect a 100-pF capacitor between pin 27 of the SPO256-AL2 and ground (pin 12 of the 24-pin connector).
c. Connect a 100-pF capacitor between pin 28 of the SPO256-AL2 and ground (pin 12 of the 24-pin connector).

STEP 9 The Audio Filter Circuit

a. Connect side 1 of a 33-kΩ resistor to pin 24 of the SPO256-AL2.
b. Connect side 2 of the first 33-kΩ resistor to side 1 of the second 33-kΩ resistor and to side 1 of the first 0.022-μF capacitor. Connect side 2 of the first 0.022-μF capacitor to ground.
c. Connect side 2 of the second 33-kΩ resistor to the positive side of the 1-μF electrolytic capacitor and to side 1 of the second 0.022-μF capacitor. Connect side 2 of the 0.022-μF capacitor to ground.
d. Connect the negative side of the 1-μF electrolytic capacitor to side 1 of the 10K potentiometer.
e. Connect side 2 (middle terminal) of the 10K potentiometer to pin 3 of the LM386.
f. Connect side 3 of the 10K potentiometer to ground.
g. Connect pins 2 and 4 of the LM386 to ground.
h. Connect a 0.1-μF capacitor between pins 4 and 7 of the LM386.
i. Connect the positive side of the 10-μF electrolytic capacitor to pin 1 of the LM386 and the negative side to pin 8 of the LM386.
j. Connect pin 6 of the LM386 to +5 V.
k. Connect a 0.1-μF capacitor between pin 6 of the LM386 and ground.
l. Connect pin 5 of the LM386 to side 1 of a 10-Ω resistor.

Connect side 2 of the 10-Ω resistor to side 1 of a 0.1-μF capacitor. Connect side 2 of the 0.1-μF capacitor to ground.

m. Connect the positive side of the 100-μF electrolytic capacitor to pin 5 of the LM386. Connect the negative side of the 100-μF capacitor to side 1 of the speaker jack. Connect side 2 of the speaker jack to ground.

STEP 10 Insertion of the Integrated Circuits

NOTE: When inserting the IC's, be sure that pin 1 of the IC lines up with pin 1 of the socket that was labeled in step 2.

a. Insert the SPO256-AL2 into the 28-pin socket.
b. Insert the LM386 into the 8-pin socket.

STEP 11 Proceed to the section on Complete Module and Operating Instructions.

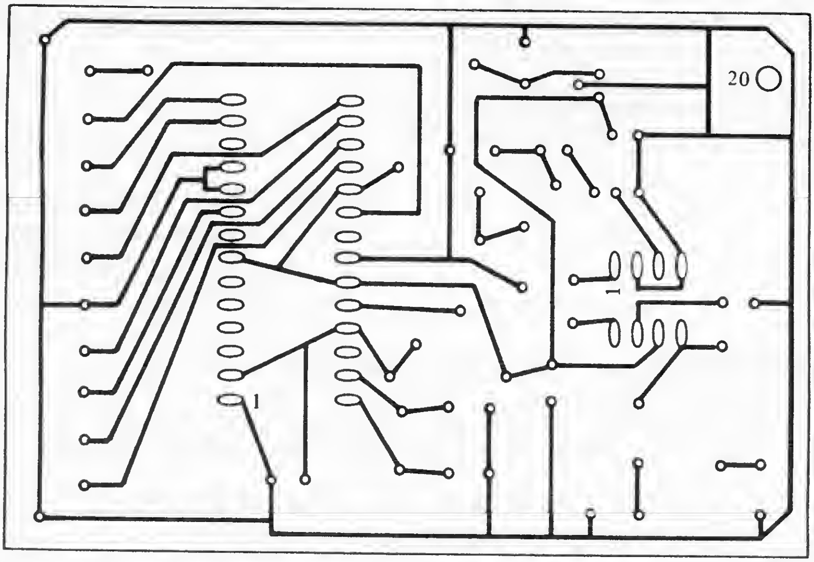
COMPLETE KIT ASSEMBLY INSTRUCTIONS

When constructing the complete kit, refer to Fig. 7-2. It illustrates the PC board and component placements. The **circuit side** of the board is the side where all the solder lines appear. (It is this side where all the components are to be soldered.) The other side (or flip side) is known as the **component side.** All the components are to be mounted or inserted into the appropriate holes on this side of the board, except where noted differently in the instructions.

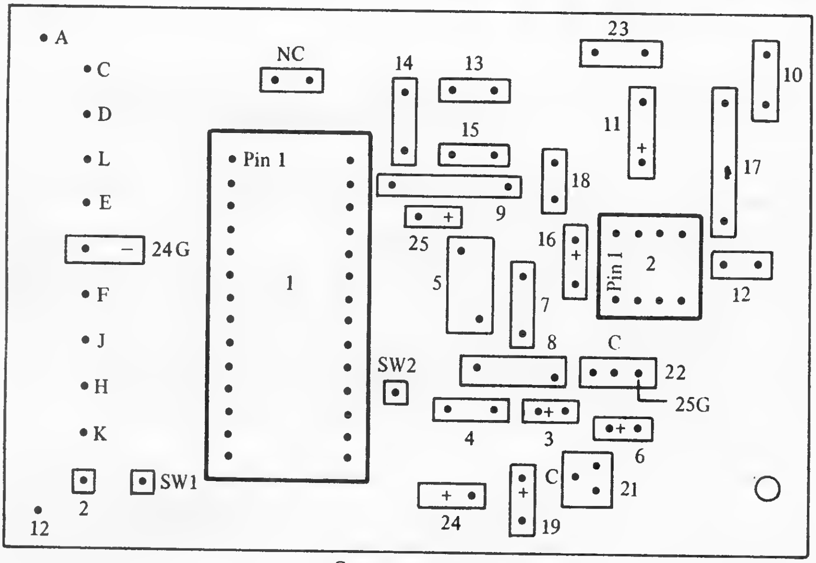
The parts for each speech synthesizer kit are shown in Table 7-1. The PC board identification of the components refers to the place on the PC board where the components should be inserted.

For example, in Table 7-1 the 33-kΩ resistors are labeled 5 and 8. If you look at Fig. 7-2, you will notice that there are two places labeled 5 and 8. These are the spots where the resistors are to be inserted.

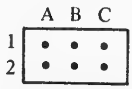
NOTE: After completing your synthesizer kit, see the section on Complete Module and Operating Instructions.



Circuit side



Component side



DPDT switch

Fig. 7-2 VIC 20 and Commodore 64 PC board speech circuit.

STEP 1 The PC Board Construction

Insert components 1 through 20 in the appropriate PC board locations denoted on the component side of the board. After insertion, solder the components onto the circuit side of the PC board.

NOTE: NC are holes that are *not* to be used.
Numbers 3, 6, 11, 16, and 19—The 1-μF, 10-μF, and 100-μF capacitors are all electrolytic. The polarities are denoted on the capacitors. The positive side is inserted in the holes on the component diagram labeled with a plus sign.

STEP 2 The Regulator

PC BOARD IDENTIFICATION NUMBER	*DESCRIPTION*
20	Ground plane
21	7805 (or LM340) 5-V regulator

This part must be inserted into the hole labeled 21 and mounted onto the circuit side of the PC board. The metal back of the regulator should be able to rest on the PC board ground plane labeled 20 (see circuit side diagram). Secure the regulator with the retaining bolt and nut.

NOTE: The connections are still soldered on the circuit side of the board. The "C" indicates the center connection.

STEP 3 The Connector Connections

The pin numbers labeled on the left-hand side of the component diagram correspond to the pin numbers of the VIC 20 and Commodore 64 bus as specified in the Commodore technical reference manual. Connect the ribbon cable from the connector to the PC board in this manner.

NOTE: Do not connect hole 24G to the ribbon cable, refer to Step 5 for this connection. Also, refer to Step 5 for the SW1 and SW2 connections.

STEP 4 Insertion of the Integrated Circuits

a. After soldering the 28-pin socket, insert the SPO256-AL2. Pin 1 of the SPO256-AL2 is denoted by the dot in the corner of the chip and is labeled on the diagram.
b. After soldering the 8-pin socket, insert the LM386. Pin 1 of the LM386 is denoted by the dot in the corner of the chip and is labeled on the diagram.

STEP 5 Case Assembly Mountings

The following components must be mounted in the case assembly as shown in Fig. 7-3. After mounting these com-

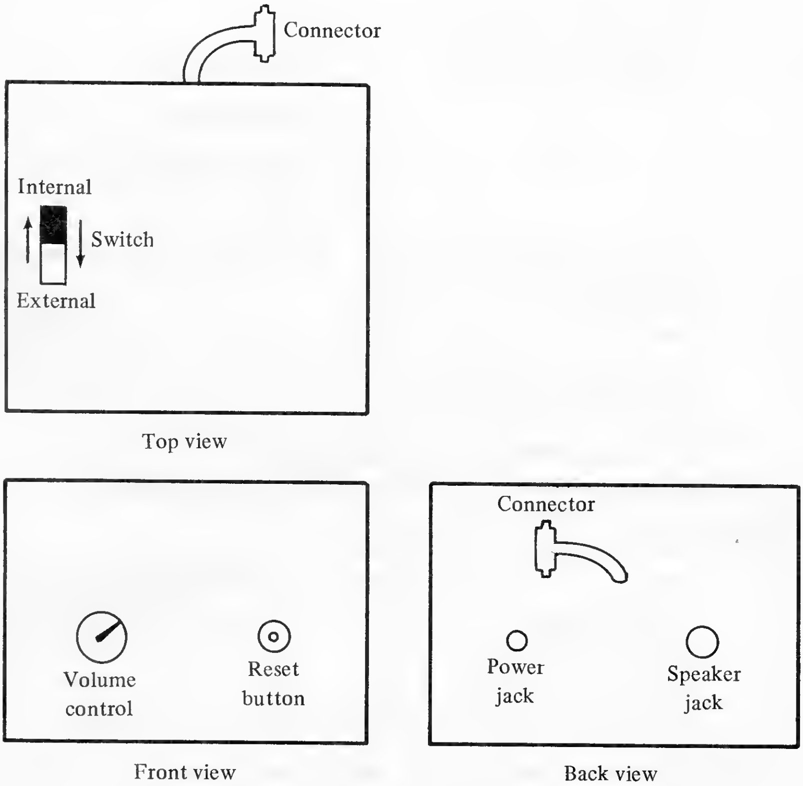


Fig. 7-3 VIC 20 and Commodore 64 component placement.

ponents into the box assembly, connect them to the following locations specified on the component diagram in Fig. 7-2.

NOTE: "C" denoted on the component diagram represents the center connection for the potentiometer.

COMPONENT TO BE MOUNTED	*CONNECT TO PC BOARD IDENTIFICATION NUMBER*	*DESCRIPTION*
10K potentiometer	22	Volume control
$\frac{1}{4}$" Speaker jack	23	Speaker connections
"Mini" power jack tip	Side 1B of the DPDT switch	Positive 9-V input power jack connection
"Mini" power jack ground	24G	Negative 9-V input power jack connection
Side 1C of the DPDT switch	24	Positive 9-V input to PC board
Reset switch (side 1)	25	Reset switch connection
Reset switch (side 2)	25G	Reset switch connection
Internal/external DPDT switch (side 2A)	SW1	Internal power connection
Internal/external DPDT switch (side 2B)	SW2	Internal power connection

STEP 6 Case Assembly

a. Secure the PC board into the box assembly with silicone sealer.

b. Install the bottom of the case onto the top of the case with the 4 holding screws.

COMPLETE MODULE AND OPERATING INSTRUCTIONS

Whether you have built the starter kit or complete kit, or purchased the complete module, you are now ready to operate your speech synthesizer (see Fig. 7-3).

Reset Button—Depressing this button readies the synthesizer for operation. It will also cause the board to stop talking.

Volume Control—Turning this control to the right or left will increase or decrease the volume of the synthesizer.

Speaker Jack—This connection will drive any 4- or 8-Ω speaker or act as an auxiliary input to any receiver.

Power Jack—Optional power connection to be used for expanded system operation only (see the section on Port Locations and Power Specifications at the beginning of this chapter). This allows you to supply the additional power required if the other modules are being used. The power supply required is a 9-V, 300-mA supply, mini-jack (the tip is positive).

CAUTION: When using external power the internal/external switch should be in the external position. This disables the internal power circuit.

Internal/External Switch—While in the internal position, the synthesizer board is powered internally from the Commodore 64 computer. This is disconnected in the external position. The VIC 20 always has to be powered externally.

24-Pin Dual Edge Connector—This connector is the interface between the synthesizer and the user port of the VIC 20 or the Commodore 64.

Operating Procedure

Using the Synthesizer with Internal Power (External Power Supply Not Required)

STEP 1 Switch the internal/external switch to the external position. This disables the internal power circuit.
STEP 2 Plug the speaker into the speaker jack.

CAUTION: The computer must be off.

STEP 3 Plug the synthesizer into the 24-pin bus (user port) of the VIC 20 or the Commodore 64.
STEP 4 Turn your computer on.
STEP 5 Turn the synthesizer switch to the internal power position.
STEP 6 Depress the reset button.

NOTE: A faint "click" should be heard in the speaker when depressing this button. If this sound is not heard, turn up the volume and try it again.

STEP 7 Skip to the section on Testing Your Circuit.

Using the Synthesizer with External Power

STEP 1 Switch the internal/external switch to the external position. This disables the internal power circuit.
STEP 2 Plug the speaker into the speaker jack.
STEP 3 Plug the external power supply into the power jack of the synthesizer board (the tip is positive).

NOTE: Do not plug into the wall outlet yet.

CAUTION: The computer must be off.

STEP 4 Plug the synthesizer into the 24-pin bus (user port) of the VIC 20 or the Commodore 64.
STEP 5 Plug in the 9-V power adapter to the synthesizer into the wall outlet.
STEP 6 Turn your computer on.
STEP 7 Depress the reset button.

NOTE: A faint "click" should be heard in the speaker when depressing this button. If this sound is not heard, turn up the volume and try it again. If the condition still exists, proceed to the following section.

TESTING YOUR CIRCUIT

After carefully wiring up your speech synthesizer board, the following commands can be used to test your circuit before loading in the respective programs. Before power is applied, visually inspect your hardware to ensure that the proper connections have been made and all the grounds are secure. On power up, a hardware reset is required—simply close the switch momentarily. A click or pop should be heard in the speaker. If this occurs, proceed with the instructions in this section. If this condition does not occur, refer to the section on Debugging Your Hardware at the end of this chapter.

VIC 20

When testing this circuit, you must first set the user port to accept data. To do so, enter the statement POKE 37136,255. Then enter the statement POKE 37138,127. This brings all the data lines including the $\overline{\text{ALD}}$ pulse high. The next statement to be entered is POKE 37136,5. This will output address 5 (allophone OY) on the data bus of the speech chip and bring $\overline{\text{ALD}}$ low. POKE 37136,69 will bring $\overline{\text{ALD}}$ high again, causing the synthesizer to talk. If everything is correct, the board will continue to speak until a pause is entered. To enter a pause and silence the board, enter POKE 37136,0 and then POKE 37136,64. This is a pause at location 000. You are now ready to load in the Exclusive Program in the usual manner (as specified in your computer manual) and create your own phrases.

Commodore 64

When testing this circuit, you must first set the user port to accept data. To do so, enter the statement POKE 56577,255. Then enter

the statement POKE 56579,127. This brings all the data lines including the $\overline{ALD}$ pulse high. The next statement to be entered is POKE 56577,5. This will output address 5 the (allophone OY) on the data bus of the speech chip and bring $\overline{ALD}$ low. POKE 56577,69 will bring $\overline{ALD}$ high again, causing the synthesizer to talk. If everything is correct, the board will continue to speak until a pause is entered. To enter a pause and silence the board, enter POKE 56577,0 and then POKE 56577,64. This is a pause at location 000. You are now ready to load the Exclusive Program in the usual manner (as specified in your computer manual) and create your own allophone phrases.

SOFTWARE

The Exclusive Phrase Finder Programs that follow allow you to build words and phrases from their constituent allophones. The phrase can be edited by moving a pointer left or right to the desired position (see Table 7-2). Inserting, deleting, or replacing allophones can then be accomplished quite easily. When the phrase is prepared to your satisfaction, a simple ENTER, NEW LINE, or RETURN will signal the synthesizer to talk.

Table 7-2 ALLOPHONE EDITING COMMANDS

AVAILABLE COMMANDS	OPERATION
"phoneme strings"	Causes named allophone to be added to the phrase at the current position of the pointer, by either replacing the existing allophone or inserting one before it (see the section on Inserting an Allophone).
"L"	Moves the position pointer left one allophone.
"R"	Moves the position pointer right one allophone.
"D"	Deletes allophone at the current position pointer.

Table 7-2 (*Continued*)

AVAILABLE COMMANDS	OPERATION
NEW LINE, ENTER, or RETURN	Causes the system to output to the hardware the commands necessary to pronounce the phrase.
"E"	Exits the program.
"XL"	Moves the position pointer "X" number of allophones to the left.
"XR"	Moves the position pointer "X" number of allophones to the right.
"I"	Turns on the "insert mode." The next allophone entered will be inserted into the phrase at the current position of the pointer. Additional allophones will be inserted until "I" is entered again. The second "I" command will turn the insert mode off. When the insert mode is off, an entered allophone will replace the one at the current position. This is the default at system start-up.

EXCLUSIVE PHRASE FINDER PROGRAM DESCRIPTION

At system start-up, a brief message will be spoken and the following commands are performed.

> NOTE: This message will be spoken each time the program is RUN. To delete this message from the system, insert the command "125 GOTO 190" into the program.

The screen is cleared and all variables are initialized. The allophone symbol array is initialized with the 64 two- or three-character symbols that represent each allophone. These are strings the user will enter in order to add an allophone to the phrase (see Tables 3-1 and 9-1). The position pointer is at position one, and the user is prompted for input with a "≥." At this prompt, an allophone

or any of the commands in Table 7-2 may be entered. An invalid allophone will be flagged as an error, as will attempting to move the position pointer to the left or right of the boundaries of the phrase. After each command, the updated phrase is displayed with the current position indicated by "⋟."

For example, at system start-up, the screen will look like this:

COMMAND	*SCREEN*	*COMMENTS*
RUN	⋟ ?	Pointer at position one.
HH1	⋟ ?HH1	Desired allophone.
ENTER	HH1⋟ ?	The first allophone has been entered; the pointer is at position two; the system is ready for the next input.
EH1	HH1⋟ ?EH1	Next allophone.
ENTER	HH1⋟ ?EH1 ***INVALID ENTRY***	User entered invalid data.
EH	HH1⋟ ?EH	
ENTER	HH1 EH⋟	The second allophone has been entered; the pointer is at position three ; the system is ready for the next input.

Editing Features

Upon entering a string of allophones and noticing that a few corrections are in order, the following edit commands are useful.

First, we must position the pointer at the location where an editing command is to be performed. Let's take the word "hello" for example. The screen should now look like this:

HH1 EH LL UW1⋗

Realizing that the UW1 allophone is incorrect, we would like to REPLACE it with "OW." To do so, the following commands are required:

Replacing an Allophone

COMMAND	*SCREEN*	*COMMENTS*
"L"	HH1 EH LL UW1⋗ L	We have to move the pointer left one space to replace UW1.
"ENTER"	HH1 EH LL⋗ UW1	The system is now ready to replace the allophone UW1.
"OW"	HH1 EH LL⋗ UW1 OW	The desired replacement allophone is typed.
"ENTER"	HH1 EH LL OW⋗	After pressing ENTER, the allophone has been replaced.

Moving the position pointer right works in the same manner as moving to the left. The only exception is that we use an "R" instead of an "L." These commands move the pointer one space at the time.

In replace mode (the default at program start-up), the new allophone will replace the allophone at the current position in the phrase.

NOTE: If you attempt to move the position pointer left or right beyond its boundaries, the message ***INVALID ENTRY*** will appear.

COMMAND	*SCREEN*	*COMMENTS*
"R"	HH1 EH LL OW≥	To clear the invalid entry, press ENTER or type in a valid command.
ENTER	R ***INVALID ENTRY***	
"L"	≥HH1 EH LL OW	
ENTER	L ***INVALID ENTRY***	

The XL and XR commands work in the same manner as the L and R commands. The only difference is that these commands move the position pointer "X" number of allophones (or spaces) to the left or right. Once again, exceeding the boundaries will prompt an ***INVALID ENTRY*** message. For example:

COMMAND	*SCREEN*	*COMMENTS*
"2L"	HH1 EH LL OW1≥	
ENTER	HH1 EH≥ LL OW	The position pointer has moved 2 spaces to the left.

Deleting an Allophone. We'll use the same examples as above. Remember, we must first position the pointer to the specified allophone to be deleted. Once this is accomplished, the following commands are required.

COMMAND	*SCREEN*	*COMMENTS*
"L" ENTER (3 times)	HH1 EH EH LL OW≥	We need to delete an EH here, so first we must move the pointer 3 spaces to the left.

COMMAND	*SCREEN*	*COMMENTS*
"D"	HH1 EH⩾ EH LL OW D	The pointer is now positioned to the allophone to be deleted.
"ENTER"	HH1 EH⩾ LL OW	The EH allophone has been deleted.

NOTE: The delete command "D" deletes one allophone at a time.

Inserting an Allophone. After creating your allophone phrases, and realizing that a few pauses (or other allophones) need to be inserted, the following command sequence must be performed. Let's use the word "chatter" as an example.

COMMAND	*SCREEN*	*COMMENTS*
"L" ENTER (3 times)	CH AE TT2 ER1 PA3⩾ ?	We must position the pointer at the location where the inserted allophone will go.
"I"	CH AE⩾ TT2 ER1 PA3 ?I	We are ready to turn on the "insert mode."
"ENTER"	CH AE⩾ TT2 ER1 PA3 ?	Insert mode has been turned on.
"PA3"	CH AE⩾ TT2 ER1 PA3 ? PA3	The desired allophone to be inserted is typed.
"ENTER"	CH AE PA3⩾ TT2 ER2 PA3 ?	The allophone has been inserted. Additional allophones may be entered at this time if required.

COMMAND	*SCREEN*	*COMMENTS*
"I"	CH AE PA3> TT2 ER2 PA3 ?I	We are ready to turn off the insert mode.
"ENTER"	CH AE PA3> TT2 ER2 PA3 ?	Insert mode has been turned off.

MAKING YOUR COMPUTER TALK

A simple "ENTER," "RETURN," or "NEW LINE" is all that's required to make the system talk. If a new allophone string is desired, you must first EXIT (E) the program and then RUN it again. This will clear all the previous allophone codes stored. Unfortunately at the time of writing this book, a "SAVE" routine was not supplied with the Exclusive Phrase Finder Program. You may want to try writing your own subroutine to save the allophone strings you've created with the program (in separate files). An alternative method is to save the Exclusive Phrase Finder Program on cassette tape or disk, with the phrase or word you've just created. When reloading the program, the following commands are necessary for it to speak the prestored phrase.

VIC 20	GOTO 4999
Commodore 64	GOTO 4999

Upon typing this command, the computer will speak the prestored allophone string and the allophone codes will be printed on the screen. The system is now ready to edit the existing string. To enter a new string, RUN the program.

EXCLUSIVE PHRASE FINDER PROGRAM LISTINGS

VIC 20

NOTE: When typing in the allophones, it is essential to type a blank space in place of all **b**'s in the program below.

```
1      FIRST = 1
5      LET LE=128
10     DIM A$(64)
20     DIM E(LE)
30     LET P=0
40     LET LN=0
100    FOR X=1 TO 64
110    READ A$(X)
120    NEXT X
130    LN=52
140    FOR X = 1 TO LN
150    READ E(X)
160    NEXT X
170    GOTO 5000
190    REM *UPDATE SREEN*
200    PRINT CHR$(147)
205    IF FIRST THEN FIRST = 0 : LN = 0
210    IF P=0 THEN 250
220    FOR X=1 TO P
230    PRINT A$(E(X)) ; "b";
240    NEXT X
250    PRINT "→"; "b";
260    IF P=LN THEN PRINT: GOTO 292
270    FOR X=P+1 TO LN
280    PRINT A$(E(X)) ; "b";
290    NEXT X
292    PRINT
295    IF M=1 THEN PRINT "***INSERT MODE***"
297    LET J=1
299    REM *INPUT COMMAND OR ALLOPHONE*
300    LET I$="": INPUT I$
310    IF I$=""  THEN 5000
320    IF I$="I" THEN  500
330    IF I$="D" THEN  600
340    IF I$="L" THEN  700
350    IF I$="R" THEN  800
360    IF I$="E" THEN  STOP
365    IF ASC(I$)<58 THEN 1000
367    REM *DECODE ALLOPHONE*
370    FOR X=1 TO 64
380    IF I$=A$(X) THEN 900
390    NEXT X
400    PRINT "***INVALID ENTRY***"
410    GOTO 300
490    REM *TOGGLE INSERT MODE*
500    LET M=ABS(M-1)
510    GOTO 200
590    REM *DELETE ALLOPHONE*
600    IF P=LN THEN 400
610    FOR X=P+1 TO LN
620    IF X=LE THEN 640
630    LET E(X)=E(X+1)
```

```
 640      NEXT X
 650      LET LN=LN-1
 660      IF P>LN THEN P=LN
 670      GOTO 200
 690      REM *MOVE LEFT*
 700      IF P-J<0 THEN 400
 710      LET P=P-J
 720      GOTO 200
 790      REM *MOVE RIGHT*
 800      IF (P+J>LN) OR (P+J>LE) THEN 400
 810      LET P=P+J
 820      GOTO 200
 890      REM *ADD ALLOPHONE TO STRING*
 900      IF LN=LE THEN 400
 910      IF M=0 THEN 960
 920      FOR Y=LN TO P+1 STEP -1
 930      LET E(Y+1) = E(Y)
 940      NEXT Y
 950      LET LN=LN+1
 960      LET P=P+1
 970      LET E(P)=X
 980      IF P>LN THEN LN=P
 990      GOTO 200
 999      REM *GET NUMBER OF SPACES TO MOVE LEFT OR RIGHT*
1000      LET J=VAL(I$)
1010      IF RIGHT$(I$,1) = "R" THEN 800
1020      IF RIGHT$(I$,1) = "L" THEN 700
1030      GOTO 400
4999      REM *SPEAK* FOR THE VIC 20
5000      POKE 37136,255
5010      POKE 37138,127
5020      FOR X=1 TO LN
5030      WAIT 37136,128,128
5040      POKE 37136, E(X)-1
5050      POKE 37136, E(X)+63
5060      NEXT X
5070      WAIT 37136,128,128
5080      POKE 37136,0
5090      POKE 37136,64
5100      GOTO 200
6999      REM *DATA*
7000      DATA PA1,PA2,PA3,PA4,PA5,OY,AY,EH,KK3,PP,
          JH,NN1,IH,TT2,RR1,AX,MM,TT1,DH1,IY,EY
7010      DATA DD1,UW1,AO,AA,YY2,AE,HH1,BB1,TH,UH,
          UW2,AW,DD2,GG3,VV,GG1,SH,ZH,RR2,FF,KK2,KK1
7020      DATA ZZ,NG,LL,WW,XR,WH,YY1,CH,ER1,ER2,
          OW,DH2,SS,NN2,HH2,OR,AR,YR,GG2,EL,BB2
9000      DATA 28,8,46,54,5,5,5,7,3,43,27,12,3,37,13,
               36,3,30,20,3,10,33,52,3
9100      DATA 16,16,36,3,56,1,10,20,1,51,3,14,32,3,
               26,59,3,43,16,17,1
9200      DATA 10,50,23,1,14,52,5
```

Commodore 64

NOTE: When typing in the allophones, it is essential to type a blank space in place of all **b**'s in the program below.

Follow the program listing for the VIC 20 except that lines 4999–5100 should be replaced with the following:

```
4999     REM *SPEAK* FOR THE COMMODORE 64
5000     POKE 56577,255
5010     POKE 56579,127
5020     FOR X=1 TO LN
5030     WAIT 56577,128,128
5040     POKE 56577, E(X)-1
5050     POKE 56577, E(X)+63
5060     NEXT X
5070     WAIT 56577,128,128
5080     POKE 56577,0
5090     POKE 56577,64
5100     GOTO 200
```

SAMPLE PROGRAM

The following program describes how to add "N" phrases to your existing program.

Data Statements

These statements must appear in the program before the lines that enable the synthesizer to speak.

NOTE: The decimal codes for each particular allophone (as shown in Table 9-1) must be inserted in the places denoted by an asterisk.

```
100   LET S$(1)=CHR$(*)+. . . .+CHR$(64)
110   LET S$(2)=CHR$(*)+. . . .+CHR$(64)
120   LET S$(3)=CHR$(*)+. . . .+CHR$(64)
130   LET S$(N)=CHR$(*)+. . . .+CHR$(64)
```

Subroutine

```
10000   B=37136 (for the VIC 20)
        B=56577 (for the Commodore 64)
10010   POKE B,255
10020   POKE B+2,127
10030   N=1
10040   S=ASC(MID$(S$(X),N,1))
10050   POKE B,S AND 63
10060   POKE B,S OR 64
10070   N=N+1
10080   IF S<64 THEN 10040
```

First Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the first phrase.

```
1000   LET N=1
1010   GOSUB 10000
```

Second Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the second phrase.

```
2000   LET N=2
2010   GOSUB 10000
```

Third Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the third phrase.

```
3000   LET N=3
3010   GOSUB 10000
```

Nth Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the Nth phrase.

```
N000   LET N=N
N010   GOSUB 10000
```

LET'S GET TECHNICAL . . .

DRIVER SUBROUTINES

This next section describes the driver, or talk subroutines, for the VIC 20 and Commodore 64. These driver subroutines can be used directly in your own application programs, or studied as examples.

VIC 20

```
5000    POKE 37136,255
5010    POKE 37138,127
5020    FOR X=1 TO LN
5030    WAIT 37136,128,128
5040    POKE 37136,E(X)−1
5050    POKE 37136,E(X)+63
5060    NEXT X
```

Explanation of the Program

Line 5000 POKE 37136,255

This sets all output ports to a logic 1.

Line 5010 POKE 37138,127

This sets the higher bit ($\overline{\text{ALD}}$) to a logic 1 and the lower bits (data lines and $\overline{\text{LRQ}}$) to a logic 0.

Line 5020 FOR X=1 TO LN

Selects first allophone.

Line 5030 WAIT 37136,128,128

The system waits for $\overline{\text{LRQ}}$ to go low in order to speak the first allophone. Other allophones cannot be entered at this time.

Line 5040 POKE 37136,E(X)−1

Pokes the first allophone onto data lines and brings $\overline{\text{ALD}}$ low.

Line 5050 POKE 37136,E(X)+63

Brings $\overline{\text{ALD}}$ high, completing the pulse and causing the system to speak first allophone.

Line 5060 NEXT X

Next allophone to be spoken.

Commodore 64

```
5000    POKE 56577,255
5010    POKE 56579,127
5020    FOR X=1 TO LN
5030    WAIT 56577,128,128
5040    POKE 56577,E(X)-1
5050    POKE 56577,E(X)+63
5060    NEXT X
```

Explanation of the Program. The Commodore 64 follows the explanation of the program for the VIC 20 except that the POKE statements differ, as listed.

DEBUGGING YOUR CIRCUIT

The following steps are simple test procedures to follow when debugging your speech synthesis circuit. Please note that additional test equipment may be required (e.g., oscilloscope, voltmeter, logic probe). In addition to the test equipment, you should be experienced with debugging electrical circuits.

Procedures to Ensure That Your Connections Are Correct

1. Check all the power and ground connections. Pins 7, 19, and 23 of the SPO256-AL2 are +5 V. Pins 2 and 22 of the SPO256-AL2 are negatively grounded.
2. Disconnect the audio filter circuit from pin 24. Turn up the volume on the LM386 to maximum. Send another audio

signal through the audio filter circuit. The output of a transistor radio is sufficient. Tune into a station where speech is playing rather than music. If the signal is heard at the speaker, this circuit is OK (it may be distorted somewhat). When resetting the chip, pin 24 should go to a logic 0 level. When the device is not speaking, a 40-kHz square wave is present on this pin.

3. Pins 2 and 25 of the SPO256-AL2 should be a logic 1 (+5 V). Pressing the reset button should change it to a logic 0 (0 V) and a small click should be heard in the speaker. If this doesn't occur and Procedures 1 and 2 are OK, the problem has to be in the 100-kΩ resistor, the push button switch, or the SPO256-AL2.
4. The signal on pin 26 should be oscillating. If this does not occur, either the crystal is bad or the capacitors connected to ground are the wrong capacitance. An oscilloscope or logic probe is required for this test.
5. Disconnect all the address lines (pins 10, 11, 13–18) from your microcomputer and then ground the pins (except pin 15). Connect pin 15 to +5 V. Disconnect the $\overline{\text{ALD}}$ (pin 20) and connect a momentary switch between it and +5 V. Upon depressing the momentary switch, allophone KK3 should be heard. If Steps 1–4 are operating properly and this step fails, the SPO256-AL2 is malfunctioning.
6. If Steps 1–5 are operating, proceed with the following. For the VIC 20, enter POKE 37136,255 and POKE 37138,127 and POKE 37136,5 and POKE 37136,69. For the Commodore 64, enter POKE 56577,255 and POKE 56579,127 and POKE 56577,5 and POKE 56577,69. Immediately on entering POKE 37136,69 (VIC 20) or POKE 56577,69 (Commodore 64), the state of the $\overline{\text{LRQ}}$ line (pin 9 of the SPO256-AL2) will change from a logic 0 to a logic 1. If this does not occur, then carefully recheck all your wiring.

CHAPTER EIGHT

MAKING YOUR SINCLAIR ZX80, ZX81, AND TIMEX/SINCLAIR 1000 TALK

The Sinclair ZX81 and ZX80 and the Timex/Sinclair 1000 were the first computers to be marketed for under $100. An amazing count of two million users grew within 6 months from the time they were first introduced. And now—these computers can speak. Timex says it best, "The Power (of Speech) is within Reach!"

This chapter consists of two sections. One describes the hardware interfaces, schematics, and parts required to make your Timex Sinclair computer talk. The other explains and provides the software required to drive the associated hardware. A detailed discussion of the program listing and its editing features is also included.

HARDWARE

This section explains all that's required to build a speech interface unit for your Sinclair and Timex/Sinclair computers. It is divided into three parts: the *Starter Kit,* the *Complete Kit,* and the *Complete Module.*

The starter kits for each particular model contain the components denoted in Table 8-1 by an asterisk. (They usually include the speech synthesis chip, the ceramic resonator, and the TTL components.) The remaining components (experimenter's breadboard,

connector, ribbon cable, resistors, capacitors, etc.) have to be purchased separately. These parts are very popular and if you don't have them around your lab, any electronics store (e.g., Radio Shack) will surely have them. The starter kit is the least expensive type of kit; however, it requires the most amount of time to build the circuit. If you are not familiar with wiring up breadboards, you may want to try the complete kit or complete module. The starter kits were designed for the more experienced hobbyist who wants to save a little extra money in exchange for a portion of his or her time.

The complete kits consist of all the parts listed in Table 8-1, including printed circuit (PC) boards, connectors, and cabinet. The speaker (and possibly the power supply) is the only part that has to be supplied. This kit is somewhat more expensive, but offers the ease of building the circuit on a PC board specifically designed for your particular computer. No wiring experience is necessary. All you need to know is how to use a soldering iron. This circuit is designed for the beginner who wants to learn how to build his or her own circuit and at the same time save a little money.

If after reading the instructions you are not interested in building a synthesizer yourself, you can purchase a complete module specifically designed for your computer. The complete modules are fully assembled and tested, and plug directly into the various computers, enabling them to speak in a matter of minutes!

> NOTE: The parts that form the starter kits, complete kits, and complete modules can be purchased from a number of distributors listed in the Parts Supplier Listing, Appendix B. With some distributors you may have to purchase all the parts separately, with others you may be able to purchase the parts in kit or module form. (Refer to this listing and note the parts that each company supplies before setting out to purchase your speech synthesis kits.)

All circuits, kit or module form, conform to certain requirements; they are discussed below.

PORT LOCATIONS AND POWER SPECIFICATIONS

The port locations used by this board are 0–15 and 64–79. Actually, only port 3 is being used, but any of these other locations will address the board. To change the port location from 0–15 or 64–79, see the section on Changing the Port Location at the end of this chapter.

There were several power supplies designed for the ZX80, ZX81, or T/S 1000. The ratings are 500 mA, 650 mA, and 1 A of power. All the power supplies are sufficient to power the computer and synthesizer board. If a 16K RAM peripheral card is used, the synthesizer board must be powered externally when using the 500 mA supply. The computer draws 250 mA, the synthesizer board draws 250 mA, and a 16K RAM card draws 150 mA. When using other peripherals, check their power requirements. If the total current draw exceeds the rating of the power supply, the synthesizer board should be powered externally.

PARTS LISTING

The list in Table 8-1 describes all parts required to build a speech synthesizer for your computer. Recall that the parts denoted by an asterisk combine to form the starter kits. All additional parts must be supplied and wired up on an experimenter's breadboard.

Table 8-1 TIMEX/SINCLAIR 1000 AND SINCLAIR ZX80 AND ZX81 PARTS LIST

QUANTITY	PC BOARD IDENTIFICATION FOR THE COMPLETE KIT	DESCRIPTION
1		PC board (3″ × 4″) or breadboard
1	28	Extender board ($\frac{3}{4}$″ × 2″)
1	29	Reset switch (red button)

Table 8-1 (*Continued*)

QUANTITY	PC BOARD IDENTIFICATION FOR THE COMPLETE KIT	DESCRIPTION
1	30	Power jack (sub-mini, black, 3-prong)
1	31	RCA phono speaker jack ($\frac{1}{4}''$)
1	32	10 K potentiometer (volume control, 3-prong)
1	28	46-pin amp connector (black)
1	27	7805 C or LM340–5 5-V regulator
1		* SPO256-AL2 speech synthesizer
1		* 74LS27 TTL logic circuit
1		* 74LS368 TTL logic circuit
1		LM386–1 operational amplifier (This is required to drive a 4- or 8-Ω speaker.)
1	1	14-pin socket (for the 74LS27)
1	2	16-pin socket (for the 74LS368)
1	3	8-pin socket (for the LM386)
1	4	28-pin socket (for the SPO256-AL2)
5	21,22,23,24,25	Jumpers (silver wires, $\frac{1}{2}''$ long)
1	26	* 3.12-MHz ceramic resonator (blue, CSA3.12MS2)
4 Resistors:		
2	5,6	33-kΩ (orange, orange, orange)

QUANTITY	PC BOARD IDENTIFICATION FOR THE COMPLETE KIT	DESCRIPTION
1	7	10-kΩ (brown, black, orange) or 100-kΩ (brown, black, yellow)
1	8	10-Ω (brown, black, black) or 11-Ω (brown, brown, black)
12 Capacitors:		
3	9,10,11	0.1 μF
1	12	25-μF (black cylinder) electrolytic
1	13	10-μF (blue cylinder) electrolytic
1	14	1-μF (silver cylinder) electrolytic
1	15	220-μF (orange cylinder, top and bottom terminals) electrolytic
1	16	100-μF (orange cylinder, bottom terminals only) electrolytic
2	17,18	0.025 μF (tan discs)
2	19,20	100 pF (110 J, tan discs)
Additional parts required (not included)		
1		7.5 to 9-V dc power supply (250–300mA)
		Optional—to be used only if external power is required. Can be obtained at Radio Shack, Catalog #2731455 or Catalog #603053.
1		Speaker—any 4- or 8-Ω speaker may be used; the choice is yours.

* The only parts included in the starter kits.

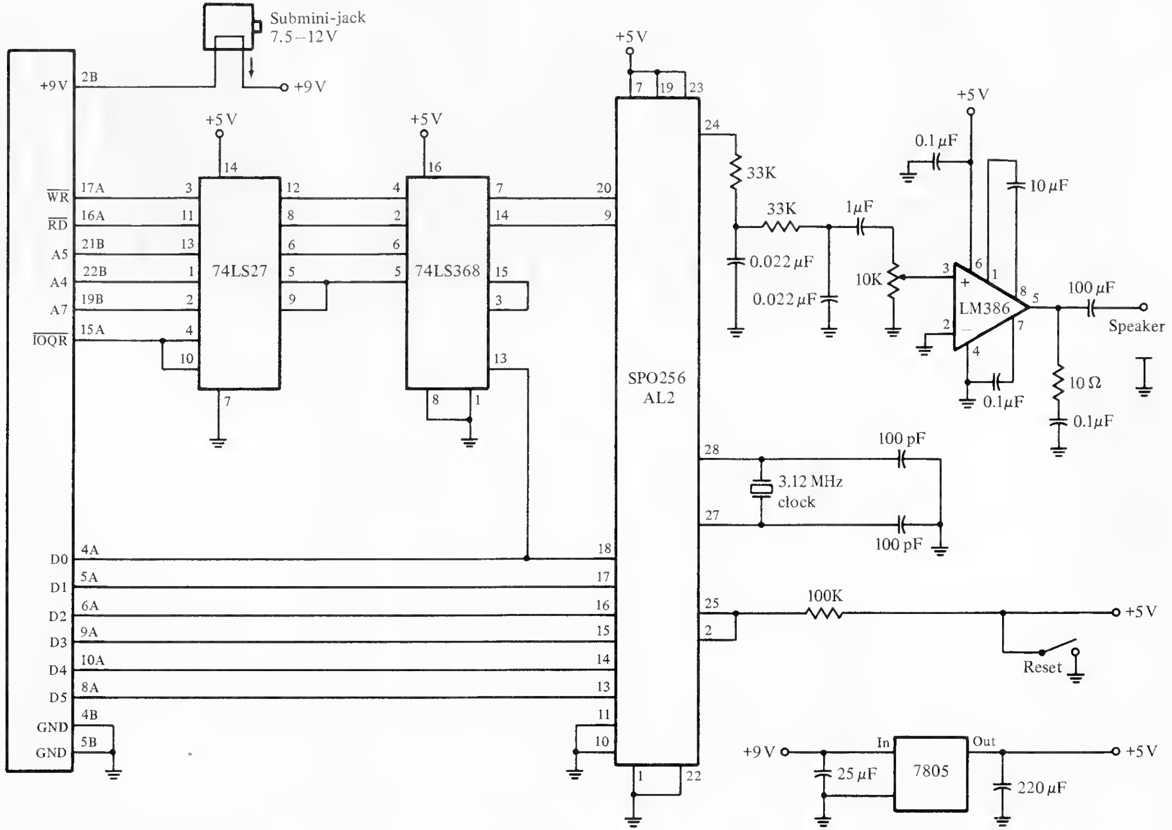


Fig. 8-1 T/S 1000, ZX80, and ZX81 speech circuit schematic.

The complete kits include all the parts in the list with the exception of a speaker and power supply. The PC board identification (included for complete kits only) describes where the parts should be placed on the PC board (see the section on Complete Kit Assembly Instructions).

STARTER KIT ASSEMBLY INSTRUCTIONS

When building the starter kit, refer to the schematic diagram in Fig. 8-1.

> NOTE: After completing your starter kit, see the section on Complete Module and Operating Instructions.

STEP 1 In addition to the starter kit, obtain all of the components listed in the Timex/Sinclair 1000 and Sinclair ZX80 and ZX81 Parts List, Table 8-1.

STEP 2 Insert the sockets into the breadboard and make a note of where pin 1 is to be located. (Pin 1 should be in the upper left-hand corner.)

STEP 3 Connect the ribbon cable to the 46-pin connector. The top row is Row A and the bottom row is Row B.

STEP 4 Solder or wirewrap the following connections:

FROM THE 46-PIN CONNECTOR (ROW A):	*TO:*
D1 pin 5A	Pin 17 of the SPO256-AL2
D2 pin 6A	Pin 16 of the SPO256-AL2
D3 pin 9A	Pin 15 of the SPO256-AL2
D4 pin 10A	Pin 14 of the SPO256-AL2
D5 pin 8A	Pin 13 of the SPO256-AL2
D0 pin 4A	Pin 18 of the SPO256-AL2 and pin 13 of the 74LS368
$\overline{\text{WR}}$ pin 17A	Pin 3 of the 74LS27
$\overline{\text{IOQR}}$ pin 15A	Pins 4 and 10 of the 74LS27
$\overline{\text{RD}}$ pin 16A	Pin 11 of the 74LS27

FROM THE 46-PIN CONNECTOR (ROW B):	*TO:*
+9 V pin 2B	Side 1 of power jack
A5 pin 21B	Pin 13 of the 74LS27
A4 pin 22B	Pin 1 of the 74LS27
A7 pin 19B	Pin 2 of the 74LS27
GND pins 4B and 5B	Common ground of the synthesizer board

STEP 5 Next solder or wirewrap these connections:

FROM THE 74LS27:	*TO:*
Pin 5	Pin 9 of the 74LS27 and pin 5 of the 74LS368
Pin 6	Pin 6 of the 74LS368
Pin 8	Pin 2 of the 74LS368
Pin 12	Pin 4 of the 74LS368

FROM THE 74LS368:	*TO:*
Pin 7	Pin 20 of the SPO256-AL2
Pin 14	Pin 9 of the SPO256-AL2
Pin 15	Pin 3 of the 74LS368

STEP 6 The Power and Ground Connections

FROM THE +5-V OUTPUT OF THE 7805 (OR LM340)	*TO:*
	Pin 14 of the 74LS27
	Pin 16 of the 74LS368
	Pins 7, 19, and 23 of the SPO256-AL2

FROM PINS 4B AND 5B OF THE 46-PIN CONNECTOR	*TO:*
	Pin 7 of the 74LS27
	Pins 1 and 8 of the 74LS368
	Pins 1 and 22 of the SPO256-AL2
	Pins 10 and 11 of the SPO256-AL2

STEP 7 The Reset Circuit

a. Connect a 100-kΩ resistor between +5 V (output of the 7805 or LM340) and pins 2 and 25 of the SPO256-AL2.
b. Connect the reset switch between ground (pin 4B of the 46-pin connector) and pins 2 and 25 of the SPO256-AL2.

STEP 8 The Regulator Circuit

a. Connect side 2 of the power jack (middle terminal) to the 9-V input side of the 7805 (or LM340).
b. Connect the negative side of the power jack to the negative terminal of the 7805 (or LM340).
c. Connect the negative terminal of the 7805 (or LM340) to pin 4B of the 46-pin connector.
d. Connect the positive side of a 25-μF electrolytic capacitor between the 9-V input of the 7805 (or LM340) and the negative side to ground (pin 4B of the 46-pin connector).
e. Connect the positive side of a 100-μF electrolytic capacitor between the 5-V output of the 7805 (or LM340) and the negative side to ground (pin 4B of the 46-pin connector).

STEP 9 The Oscillator Circuit

a. Connect the 3.12-MHz ceramic resonator between pins 27 and 28 of the SPO256-AL2.
b. Connect a 100-pF capacitor between pin 27 of the

SPO256-AL2 and ground (pin 4B of the 46-pin connector).

c. Connect a 100-pF capacitor between pin 28 of the SPO256-AL2 and ground (pin 4B of the 46-pin connector).

STEP 10 The Audio Filter Circuit

a. Connect side 1 of a 33-kΩ resistor to pin 24 of the SPO256-AL2.

b. Connect side 2 of the first 33-kΩ resistor to side 1 of the second 33-kΩ resistor and to side 1 of the first 0.022-μF capacitor. Connect side 2 of the first 0.022-μF capacitor to ground.

c. Connect side 2 of the second 33-kΩ resistor to the positive side of the 1-μF electrolytic capacitor and to side 1 of the second 0.022-μF capacitor. Connect side 2 of the 0.022-μF capacitor to ground.

d. Connect the negative side of the 1-μF electrolytic capacitor to side 1 of the 10K potentiometer.

e. Connect side 2 (middle terminal) of the 10K potentiometer to pin 3 of the LM386.

f. Connect side 3 of the 10K potentiometer to ground.

g. Connect pins 2 and 4 of the LM386 to ground.

h. Connect a 0.1-μF capacitor between pins 4 and 7 of the LM386.

i. Connect the positive side of the 10-μF electrolytic capacitor to pin 1 of the LM386 and the negative side to pin 8 of the LM386.

j. Connect pin 6 of the LM386 to +5 V.

k. Connect a 0.1-μF capacitor between pin 6 of the LM386 and ground.

l. Connect pin 5 of the LM386 to side 1 of a 10-Ω resistor. Connect side 2 of the 10-Ω resistor to side 1 of a 0.1-μF capacitor. Connect side 2 of the 0.1-μF capacitor to ground.

m. Connect the positive side of the 100-μF electrolytic capacitor to pin 5 of the LM386. Connect the negative side of the 100-μF capacitor to side 1 of the speaker jack. Connect side 2 of the speaker jack to ground.

STEP 11 Insertion of the Integrated Circuits

NOTE: When inserting the IC's, be sure that pin 1 of the IC lines up with pin 1 of the socket that was previously labeled.

a. Insert the SPO256-AL2 into the 28-pin socket.
b. Insert the 74LS27 into the 14-pin socket.
c. Insert the 74LS368 into the 16-pin socket.
d. Insert the LM386 into the 8-pin socket.

STEP 12 Skip to the section on Complete Module and Operating Instructions.

COMPLETE KIT ASSEMBLY INSTRUCTIONS

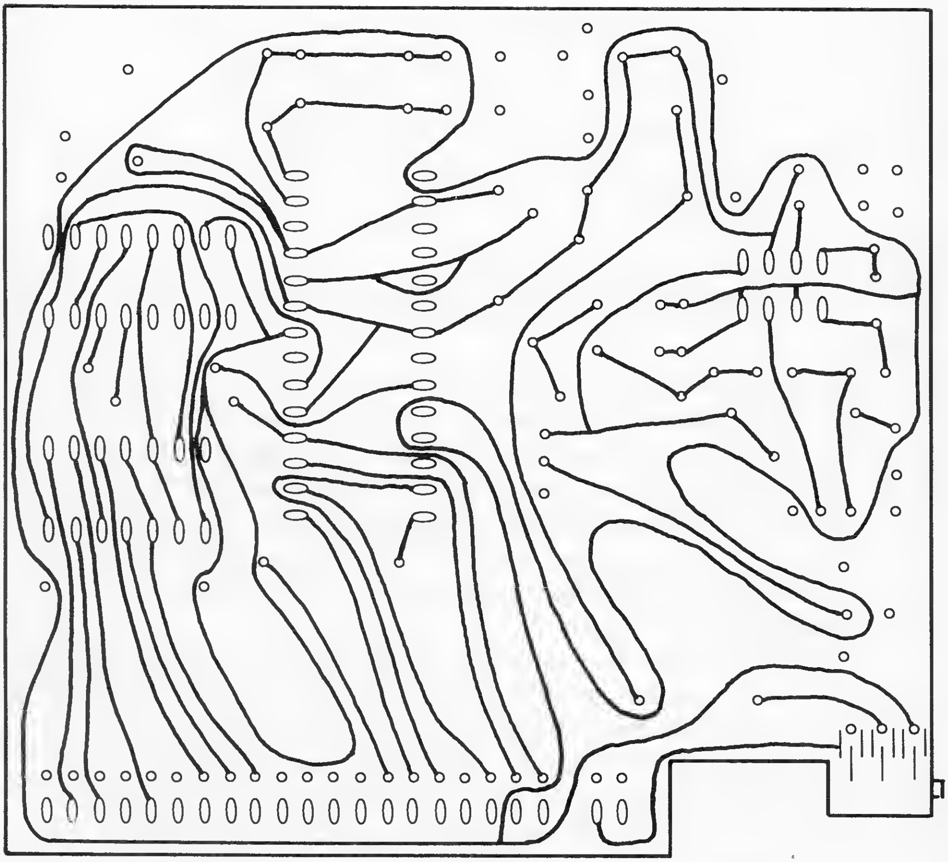
When constructing the complete kit, refer to Fig. 8-2. It illustrates the PC board and component placements. The **circuit side** of the board is the side where all the solder lines appear. (It is this side where all the components are to be soldered.) The other side (or flip side) is known as the **component side.** All the components are to be mounted or inserted into the appropriate holes on this side of the board, except where noted differently in the instructions.

The parts for each speech synthesizer kit are shown in Table 8-1. The PC board identification of the components refers to the place on the PC board where the components should be inserted.

For example, in Table 8-1 the jumpers are labeled 21, 22, 23, 24, and 25. If you look at Fig. 8-2, you will notice that there are five places labeled with those numbers. These are the holes where the jumpers are to be inserted.

NOTE: After completing your synthesizer kit, see the section on Complete Module and Operating Instructions.

STEP 1 The PC Board Construction
Insert components 1 through 26 in the appropriate PC board locations denoted on the component side of the board. After insertion, solder the components onto the circuit side of the PC board.



Circuit side

Fig. 8-2 T/S 1000, ZX80, and ZX81 PC board speech circuit.

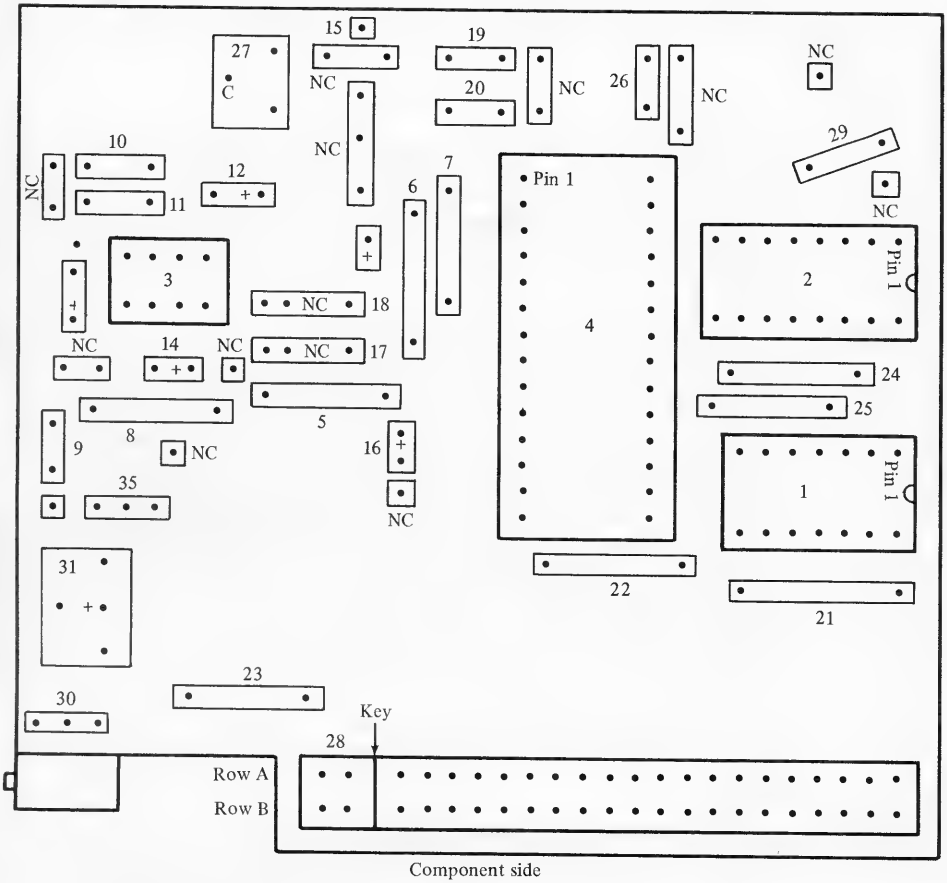


Fig. 8-2 *(Continued)*

NOTE: NC refers to holes that are *not* to be used.

NOTE: Numbers 12, 13, 14, 15, and 16—the 1-μF, 10-μF, 25-μF, 100-μF, and 220-μF capacitors are all electrolytic. The polarities are denoted on the capacitors. The positive side is inserted in the holes on the component diagram labeled with a plus sign.

STEP 2 The Regulator

When mounting, turn the board to the component side. Insert the three prongs into holes labeled 27. The "C" indicates the center connection. Make certain that the metal back of the regulator is facing the left side of the board.

STEP 3 The Connector and Extender Board

These parts must be mounted simultaneously. After inserting the connector into location 28, place the extender board inbetween the two rows of pins on the connector on the circuit side of the board. The key in the extender board must line up with the key in the connector (pin 3). Place the extender board on the upper row of pins (Row A) and line up the soldering connections. Solder the pins to the extender board. Make sure you put enough solder to fill the holes for a secure connection without shorting out the lines. To solder the bottom row, bend the pins up until they touch the extender board. Again, solder the pins onto the extender board, filling the holes on the PC board.

STEP 4 Insertion of the Integrated Circuits

a. After soldering the 28-pin socket, insert the SPO256-AL2. Pin 1 of the SPO256-AL2 is denoted by the dot in the corner of the chip and is labeled on the component diagram.
b. After soldering the 8-pin socket, insert the LM386. Pin 1 of the LM386 is denoted by the dot in the corner of the chip and is labeled on the component diagram.
c. After soldering the 16-pin socket, insert the 74LS368. Pin 1 of the 74LS368 is denoted by either a dot in the

corner of the chip, or a half circle depression or notch on the pin 1 end (see Fig. 8-2, numbers 1 and 2).

d. After soldering the 14-pin socket, insert the 74LS27. Pin 1 of the 74LS27 is denoted by a gash in the middle of the chip on the pin 1 side.

STEP 5 The Remaining Parts

The following components must be mounted on the circuit side of the PC board as shown in Figs. 8-2 and 8-3.

COMPONENT TO BE MOUNTED	*CONNECT TO PC BOARD IDENTIFICATION NUMBER*	*COMMENTS*
Reset switch	29	When mounting, line up posts of the switch with the holes in 29. The posts will not go through the holes as is. Melt a little solder in the holes to secure the switch.
Power jack	30	The 3 prongs of the jack are placed on top of the 3 solder lines in the lower right-hand corner of the circuit side of the PC board (30). The sub-mini-jack is facing away from the board,

COMPONENT TO BE MOUNTED	*CONNECT TO PC BOARD IDENTIFICATION NUMBER*	*COMMENTS*
		so a power-supply jack can be inserted into it. Make a good solder connection on all 3 prongs for a secure connection. *Note:* the tip is positive.
Phono speaker jack	31	This mounts in the four holes in box 31. The jack also faces away from the board allowing a speaker jack to be plugged into it.
Volume control	32	The "C" in the component diagram is the center connection. The arrow should be able to point up; if the arrow faces down, the volume control is in backwards.

STEP 6 The complete kit for the T/S 1000, ZX80, and ZX81 is now complete. The plastic box for the unit is not included.

COMPLETE MODULE AND OPERATING INSTRUCTIONS

Whether you have built the starter kit, complete kit or purchased the complete module, you are now ready to operate your speech synthesizer.

The speech synthesizer for your T/S 1000, ZX81, or ZX80 adapts very easily to your computer and incorporates many features. Refer to Fig. 8-3 for a description of these features.

Reset Button—Depressing this button readies the synthesizer for operation. It will also cause the board to stop talking.

Volume Control—Turning this control to the right or left will increase or decrease the volume of the synthesizer.

Speaker Jack—This connection will drive any 4- or 8-Ω speaker or act as an auxiliary input to any receiver.

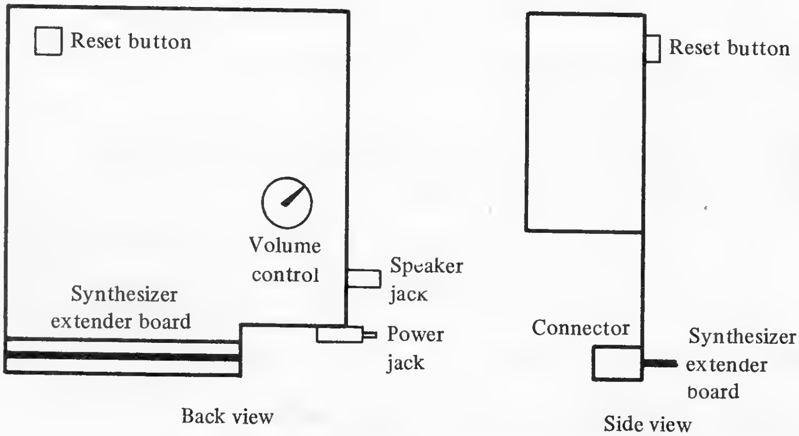


Fig. 8-3 T/S 1000, ZX80, and ZX81 component placement.

Power Jack—Optional power connection to be used for expanded system operation only (see the section on Port Locations and Power Specifications at the beginning of this chapter). This allows you to supply the additional power required if the other modules are being used. The power supply required is a 9-V, 300-mA supply, *sub-mini*-jack (the tip is positive). Upon plugging in the power supply, the internal power is automatically disconnected.

"Piggy-Back" Expandability—Other modules (e.g., RAM packs) can be plugged in behind the synthesizer.

46-Dual Edge Connector—This connector is the interface between the synthesizer and the ZX80, ZX81, or T/S 1000.

Operating Procedure

Using the Synthesizer without Expanded System Operation (No Additional Memory Packs)

NOTE: An external power supply is not required.

STEP 1 Plug the speaker into the speaker jack.

CAUTION: The computer must be off.

STEP 2 Plug the synthesizer into the 46-pin bus of the T/S 1000, ZX80, or ZX31.
STEP 3 Plug in the 9-V power adapter into your computer.
STEP 4 Plug in the 9-V power adapter into the wall outlet.
STEP 5 Depress the reset button.

NOTE: A faint "click" should be heard in the speaker when depressing this button. If this sound is not heard, turn up the volume and try it again.

STEP 6 Skip to the section on Testing Your Circuit.

Using the Synthesizer with RAM Packs

NOTE: An external power supply is required for this configuration.

STEP 1 Plug the speaker into the speaker jack.

STEP 2 Plug the external power supply into the power jack of the synthesizer (the tip is positive).

NOTE: This will disable the power being drawn from your computer. Do not plug into wall outlet yet.

CAUTION: The computer must be off.

STEP 3 Plug the synthesizer into the 46-pin bus of the T/S 1000, ZX80, or ZX81.

STEP 4 Plug the RAM into the synthesizer extender board. Check to be sure that the RAM pack is stationary.

STEP 5 Plug in the 9-V power adapter into your computer.

STEP 6 Plug in the 9-V power adapter into the wall outlet.

STEP 7 Plug in the 9-V power adapter to the synthesizer into the wall outlet.

STEP 8 Depress the reset button.

NOTE: A faint "click" should be heard in the speaker when depressing this button. If this sound is not heard, turn up the volume and try it again.

NOTE: When using the speech synthesizer with an expanded system, be certain that all modules are secure. Sinclair's design does not allow for the added weight on the back of the computer caused by the expanded system. The addition of this weight causes the front of the computer to rise. Therefore, when entering data through the keyboard, the unsteadiness may cause the computer to "crash" (i.e., sound like an untuned radio station). If so, place a piece of material (such as foam rubber) under the RAM pack to level it.

TESTING YOUR CIRCUIT

After carefully wiring up your speech synthesizer board, the following commands can be used to test your circuit before loading in the respective programs. Before power is applied, visually inspect your hardware to ensure that the proper connections have been

made and all the grounds are secure. On power up, a hardware reset is required—simply close the switch momentarily. A click or pop should be heard in the speaker. If this occurs, proceed with the instructions in this section. If this condition does not occur, refer to the section on Debugging Your Hardware at the end of this chapter.

T/S 1000 and ZX81

When testing this circuit, you must first enter the Machine Code Program listed in this chapter.

After doing so, enter the statement POKE 16522,5. This will output address 5 (the OY allophone) into the Machine Code Program at location 16522. The statement LET L=USR (16514) runs the Machine Code Program, causing the synthesizer to speak. If everything is correct, the synthesizer will continue to speak until a pause is entered. To enter a pause and silence the board, enter the following statements:

POKE 16522,0	This will output a pause 000 into the Machine Code Program.
LET L=USR (16514)	This will run the Machine Code Program.

If you are not using external memory, you are now ready to load in the Test Program in the usual manner (as specified in your computer manual) and create your own phrases. If you are using external memory, you are now ready to load in either the Exclusive Program or the Test Program.

ZX80

When testing this circuit, you must first enter the Machine Code Program listed in this chapter. After doing so, enter the statement POKE 16435,5. This will output address 5 (the OY allophone) into the Machine Code Program at location 16435. The statement LET L=USR (16427) runs the Machine Code Program, causing the synthesizer to speak. If everything is correct, the synthesizer will

continue to speak until a pause is entered. To enter a pause and silence the board, enter the following statements:

POKE 16435,0	This will output a pause 000 into the Machine Code Program.
LET L = USR (16427)	This will run the Machine Code Program.

If you are not using external memory, you are now ready to load in the Test Program in the usual manner (as specified in your computer manual) and create your own phrases. If you are using external memory, you are now ready to load in either the Exclusive program or the Test Program.

SOFTWARE

The Exclusive Phrase Finder Programs that follow allow you to build words and phrases from their constituent allophones. The phrase can be edited by moving a pointer left or right to the desired position (see Table 8-2). Inserting, deleting, or replacing allophones

Table 8-2 ALLOPHONE EDITING COMMANDS

AVAILABLE COMMANDS	OPERATION
"phoneme strings"	Causes named allophone to be added to the phrase at the current position of the pointer, by either replacing the existing allophone or inserting one before it (see the section on Inserting an Allophone).
"L"	Moves the position pointer left one allophone.
"R"	Moves the position pointer right one allophone.
"D"	Deletes allophone at the current position pointer.

Table 8-2 (*Continued*)

AVAILABLE COMMANDS	OPERATION
"I"	Turns on the "insert mode." The next allophone entered will be inserted into the phrase at the current position of the pointer. Additional allophones will be inserted until "I" is entered again. The second "I" command will turn the insert mode off. When the insert mode is off, an entered allophone will replace the one at the current position. This is the default at system start-up.
NEW LINE, ENTER, or RETURN	Causes the system to output to the hardware the commands necessary to pronounce the phrase.
"E"	Exits the program.
For the T/S 1000 and ZX81:	
"L,X"	Moves the position pointer "X" number of allophones to the left.
"R,X"	Moves the position pointer "X" number of allophones to the right.
For the ZX80:	
"L,X and R,X"	Commands are not applicable.

can then be accomplished quite easily. When the phrase is prepared to your satisfaction, a simple ENTER, NEW LINE, or RETURN will signal the synthesizer to talk.

EXCLUSIVE PHRASE FINDER PROGRAM DESCRIPTION

At system start-up, the following commands are performed. The screen is cleared and all variables are initialized. The allophone symbol array is initialized with the 64 two- or three-character

symbols that represent each allophone. These are strings the user will enter in order to add an allophone to the phrase (see Tables 3-1 and 9-1). The position pointer is at position one, and the user is prompted for input with a "⩾." At this prompt, an allophone or any of the commands in Table 8-2 may be entered. An invalid allophone will be flagged as an error, as will attempting to move the position pointer to the left or right of the boundaries of the phrase. After each command, the updated phrase is displayed with the current position indicated by "⩾."

For example, at system start-up, the screen will look like this:

COMMAND	*SCREEN*	*COMMENTS*
RUN	⩾ ?	Pointer at position one.
HH1	⩾ ?HH1	Desired allophone.
ENTER	HH1⩾ ?	The first allophone has been entered; the pointer is at position two; the system is ready for the next input.
EH1	HH1⩾ ?EH1	Next allophone.
ENTER	HH1⩾ ?EH1 ***INVALID ENTRY***	User entered invalid data.
EH	HH1⩾ ?EH	
ENTER	HH1 EH⩾	The second allophone has been entered; the pointer is at position three; the system is ready for the next input.

Editing Features

Upon entering a string of allophones and noticing that a few corrections are in order, the following edit commands are useful.

First, we must position the pointer at the location where an editing command is to be performed. Let's take the word "hello" for example. The screen should now look like this:

HH1 EH LL UW1≥

Realizing that the UW1 allophone is incorrect, we would like to REPLACE it with "OW." To do so, the following commands are required:

Replacing an Allophone

COMMAND	*SCREEN*	*COMMENTS*
"L"	HH1 EH LL UW1≥ L	We have to move the pointer left one space to replace UW1.
"ENTER"	HH1 EH LL≥ UW1	The system is now ready to replace the allophone UW1.
"OW"	HH1 EH LL≥ UW1 OW	The desired replacement allophone is typed.
"ENTER"	HH1 EH LL OW≥	After pressing ENTER, the allophone has been replaced.

Moving the position pointer right works in the same manner as moving to the left. The only exception is that we use an "R"

instead of an "L." These commands move the pointer one space at the time.

In replace mode (the default at program start-up), the new allophone will replace the allophone at the current position in the phrase.

> NOTE: If you attempt to move the position pointer left or right beyond its boundaries, the message ***INVALID ENTRY*** will appear.

COMMAND	*SCREEN*	*COMMENTS*
"R" ENTER	HH1 EH LL OW> R ***INVALID ENTRY***	To clear the invalid entry, press ENTER or type in a valid command.
"L" ENTER	>HH1 EH LL OW L ***INVALID ENTRY***	

The "L,X" and "R,X" commands work in the same manner as the L and R commands. The only difference is that these commands move the position pointer "X" number of allophones (or spaces) to the left or right. Once again, exceeding the boundaries will prompt an ***INVALID ENTRY*** message. For example:

COMMAND	*SCREEN*	*COMMENTS*
"L,2"	HH1 EH LL OW1>	
ENTER	HH1 EH> LL OW	The position pointer has moved 2 spaces to the left.

Deleting an Allophone. We'll use the same examples as above. Remember, we must first position the pointer to the specified allophone to be deleted. Once this is accomplished, the following commands are required.

COMMAND	*SCREEN*	*COMMENTS*
"L" ENTER (3 times)	HH1 EH EH LL OW>	We need to delete an EH here, so first we must move the pointer 3 spaces to the left.
"D"	HH1 EH> EH LL OW D	The pointer is now positioned to the allophone to be deleted.
"ENTER"	HH1 EH> LL OW	The EH allophone has been deleted.

NOTE: The delete command "D" deletes one allophone at a time.

Inserting an Allophone. After creating your allophone phrases, and realizing that a few pauses (or other allophones) need to be inserted, the following command sequence must be performed. Let's use the word "chatter" as an example.

COMMAND	*SCREEN*	*COMMENTS*
"L" ENTER (3 times)	CH AE TT2 ER1 PA3> ?	We must position the pointer at the location where the inserted allophone will go.
"I"	CH AE> TT2 ER1 PA3 ?I	We are ready to turn on the "insert mode."
"ENTER"	CH AE> TT2 ER1 PA3 ?	Insert mode has been turned on.
"PA3"	CH AE> TT2 ER1 PA3 ? PA3	The desired allophone to be inserted is typed.

COMMAND	*SCREEN*	*COMMENTS*
"ENTER"	CH AE PA3> TT2 ER2 PA3 ?	The allophone has been inserted. Additional allophones may be entered at this time if required.
"I"	CH AE PA3> TT2 ER2 PA3 ?I	We are ready to turn off the insert mode.
"ENTER"	CH AE PA3> TT2 ER2 PA3 ?	Insert mode has been turned off.

MAKING YOUR COMPUTER TALK

A simple "ENTER," "RETURN," or "NEW LINE" is all that's required to make the system talk. If a new allophone string is desired, you must first EXIT (E) the program and then RUN it again. This will clear all the previous allophone codes stored. Unfortunately at the time of writing this book, a "SAVE" routine was not supplied with the Exclusive Phrase Finder Program. You may want to try writing your own subroutine to save the allophone strings you've created with the program (in separate files). An alternative method is to save the Exclusive Phrase Finder Program on cassette tape or disk, with the phrase or word you've just created. When reloading the program, the following commands are necessary for it to speak the prestored phrase.

ZX80	GOTO 3000
T/S 1000 and ZX81	GOTO 3000

Upon typing this command, the computer will speak the prestored allophone string and the allophone codes will be printed on the

screen. The system is now ready to edit the existing string. To enter a new string, RUN the program.

MACHINE CODE PROGRAMS

The following driver programs must be entered into your computer before entering the Exclusive Phrase Finder Program, the Test Program, or any other program.

T/S 1000 and ZX81

STEP 1 Enter in the following program lines:

```
10    REM 012345678901:Reserves 12 bytes of memory for the
      machine code.
20    FOR I=16514 TO 16525
30    INPUT X
40    POKE I,X
50    NEXT I
```

STEP 2 RUN the program.
STEP 3 The screen will be cleared and the following machine codes must be entered one at a time:

```
219   "ENTER"
  3   "ENTER"
230   "ENTER"
  1   "ENTER"
202   "ENTER"
130   "ENTER"
 64   "ENTER"
 62   "ENTER"
  0   "ENTER"
211   "ENTER"
  3   "ENTER"
201   "ENTER"
```

STEP 4 The system is now ready for either the Test Program, the Exclusive Program, or any program you have created. When entering the new program, lines 20, 30, 40, and 50 should be replaced as shown in the Exclusive Program.

NOTE: After entering one of the above programs in Step 4, the entire source code may be saved on cassette tape. When reloading the program from the cassette tape, the Machine Code Program need not be re-entered.

ZX80

STEP 1 Enter in the following program lines:

```
10    REM 012345678901:Reserves 12 bytes of memory for the
      machine code.
20    FOR I=16427 TO 16438
30    INPUT X
40    POKE I,X
50    NEXT I
```

STEP 2 RUN the program.

STEP 3 The screen will be cleared and the following machine codes must be entered one at a time:

```
219   "ENTER"
  3   "ENTER"
230   "ENTER"
  1   "ENTER"
202   "ENTER"
 43   "ENTER"
 64   "ENTER"
 62   "ENTER"
  0   "ENTER"
211   "ENTER"
  3   "ENTER"
201   "ENTER"
```

STEP 4 The system is now ready for either the Test Program, the Exclusive Program, or any program you have created. When entering the new program, lines 20, 30, 40, and 50 should be replaced as shown in the Exclusive Program.

NOTE: After entering one of the above programs in Step 4, the entire source code may be saved on cassette tape. When reloading the program from the cassette tape, the Machine Code Program need not be re-entered.

EXCLUSIVE PHRASE FINDER PROGRAM LISTINGS

T/S 1000 and ZX81

NOTE: When typing in the allophones, it is essential to type blank spaces in place of all **b**'s in the program below. (The computer will only read allophones in groups of three letters or spaces.)

This program is to be run in the fast mode and used with the 16K expansion RAM.

```
 20     DIM A$(64*4) : REM Allophone Array
 30     DIM Z$(64,3)
 35     LET LENGTH = 128 : REM CHANGING 128 WILL CHANGE
        THE MAXIMUM LENGTH OF THE STRING THAT CAN BE
        ENTERED AT ANY ONE TIME
 40     DIM S(LENGTH)
 50     LET A$ = "PA1,PA2,PA3,PA4,PA5,OYb,AYb,EHb,KK3,
        PPb,JHb,NN1,IHb,TT2,RR1,AXb,MMb,TT1,DH1,IYb,EYb,
        DD1,UW1,AOb,AAb,YY2,AEb,HH1,BB1,THb,UHb,UW2,
        AWb,DD2,GG3,VVb,GG1,SHb,ZHb,RR2,FFb,KK2,KK1,
        ZZb,NGb,LLb,WWb,XRb,WHb,YY1,CHb,ER1,ER2,OWb,
        DH2,SSb,NN2,HH2,ORb,ARb,YRb,GG2,ELb,BB2"
 60     FOR I=1 TO 64
 70     FOR J=1 TO 3
 80     LET Z$(I,J)=A$((I-1)*4+J)
 90     NEXT J
 95     NEXT I
160     LET L=O : REM LENGTH OF PHRASE (IN ALLOPHONES)
170     LET P=1 : REM CURRENT POSITION IN PHRASE
180     LET M=0 : REM MODE FLAG (0=REPLACE MODE,
        1=INSERT MODE)
190     LET B=16514 : REM MACHINE CODE ADDRESS
```

```
290     REM MAIN PROGRAM LOOP:
300     GOSUB 2000: REM MACHINE CODE ADDRESS
        DISPLAY CURRENT PHRASE
315     INPUT I$
317     LET LN=1
320     IF I$ = ""  THEN GOTO 3000 : REM TALK
330     IF I$ = "E" THEN STOP : REM EXIT
340     IF I$ = "L" THEN GOTO 400 : REM MOVE LEFT
350     IF I$ = "R" THEN GOTO 500 : REM MOVE RIGHT
360     IF I$ = "D" THEN GOTO 600 : REM DELETE ALLOPHONE
370     IF I$ = "I" THEN GOTO 700 : REM TURN ON INSERT MODE
372     IF LEN I$ = 1 THEN GOTO 900
375     IF I$(2) = "," THEN GOTO 5000
380     GOTO 800
400     REM MOVE LEFT
410     IF P<LN+1 THEN GOTO 900
420     LET P=P-LN
430     GOTO 300
500     REM MOVE RIGHT
510     IF (P>L-LN+1) OR (P>LENGTH - 1) THEN GOTO 900
530     LET P=P+LN
540     GOTO 300
600     REM DELETE ALLOPHONE
610     IF P>L THEN GOTO 300
620     FOR I=P TO L
630     LET S(I) = S(I+1)
640     NEXT I
650     LET L = L-1
660     GOTO 300
700     REM TURN ON INSERT MODE
710     LET M = ABS(M-1)
720     GOTO 300
800     REM TEST FOR VALID ALLOPHONE ENTRY
805     IF LEN(I$)=2 THEN LET I$ = I$ + "b"
810     FOR I = 1 TO 64
830     IF I$ = Z$(I) THEN GOTO 950
840     NEXT I
900     REM USER ENTERED INVALID DATA
910     PRINT "***INVALID ENTRY***"
920     GOTO 315 : REM RETURN TO MAINLINE, DON'T
        REFRESH SCREEN
950     REM USER ENTERED VALID ALLOPHONE
960     IF P>LENGTH THEN GOTO 900
970     IF M=0 THEN GOTO 1100
980     IF L>LENGTH -1 THEN GOTO 900 : REM ERROR-NO
        ROOM TO INSERT
990     LET X = S(P)
1000    FOR J = P+1 TO L+1
1010    LET Y = S(J)
1020    LET S(J) = X
1030    LET X=Y
1040    NEXT J
1050    LET L = L+1
1100    REM NOW PUT ALLOPHONE INTO STRING AT POS P
1110    LET S(P) = 1
1120    IF L<P THEN LET L = P
1130    LET P = P+1
```

```
1150      GOTO 300
2000      REM DISPLAY ALLOPHONE STRING
2010      CLS
2020      PRINT,,,,,,
2030      IF P=1 THEN PRINT "->";
2040      FOR I = 1 TO L
2060      PRINT Z$(S(I));"b";
2070      IF I=P-1 THEN PRINT "->";
2080      NEXT I
2090      PRINT
2105      IF M=1 THEN PRINT "***INSERT MODE***"
2110      RETURN
3000      REM SECTION TO MAKE BOARD PRONOUNCE PHRASE
3020      FOR I=1 TO L
3050      POKE(B+8), S(I)-1 : REM POKE ALLOPHONE
          INTO MACHINE CODE
3060      LET R=USR(B) : REM PRONOUNCE ALLOPHONE
3070      NEXT I
3080      POKE(B+B), O
3090      LET R=USR(B)
3100      GOTO 300
4000      STOP
5000      LET LN = VAL I$(3 TO)
5010      IF I$(1) = "L" THEN GOTO 400
5020      IF I$(1) = "R" THEN GOTO 500
5030      GOTO 900
```

ZX80

NOTE: When typing in the allophones, it is essential to type blank spaces in place of all **b**'s in the program below. (The computer will only read allophones in groups of three letters or spaces.)

This program is to be used with the 16K expansion RAM.

```
30        DIM Z(64*3)
35        LET LENGTH = 128 : REM CHANGING 128 WILL CHANGE
          THE MAXIMUM LENGTH OF THE STRING THAT CAN BE
          ENTERED AT ANY ONE TIME
40        DIM S(LENGTH)
50        LET A$ = "PA1,PA2,PA3,PA4,PA5,OYb,AYb,EHb,KK3,
          PPb,JHb,NN1,IHb,TT2,RR1,AXb,MMb,TT1,DH1,IYb,EYb,
          DD1,UW1,AOb,AAb,YY2,AEb,HH1,BB1,THb,UHb,UW2,
          AWb,DD2,GG3,VVb,GG1,SHb,ZHb,RR2,FFb,KK2,KK1,
          ZZb,NGb,LLb,WWb,XRb,WHb,YY1,CHb,ER1,ER2,OWb,
          DH2,SSb,NN2,HH2,ORb,ARb,YRb,GG2,ELb,BB2"
60        FOR I=1 TO 64
70        FOR J=1 TO 3
80        LET Z((I-1)*3+J)=CODE(A$)
85        LET A$ = TL$(A$)
```

```
 90     NEXT J
 92     LET A$ = TL$(A$)
 95     NEXT I
160     LET L=0 : REM LENGTH OF PHRASE (IN ALLOPHONES)
170     LET P=1 : REM CURRENT POSITION IN PHRASE
180     LET M=0 : REM MODE FLAG (0=REPLACE MODE,
        1=INSERT MODE)
190     LET B=16427 : REM MACHINE CODE ADDRESS
290     REM MAIN PROGRAM LOOP:
300     GOSUB 2000 : REM MAIN PROGRAM LOOP BEGINNING.
        DISPLAY CURRENT PHRASE
315     INPUT I$
320     IF I$ = ""  THEN GOTO 3000 : REM TALK
330     IF I$ = "E" THEN STOP : REM EXIT
340     IF I$ = "L" THEN GOTO 400 : REM MOVE LEFT
350     IF I$ = "R" THEN GOTO 500 : REM MOVE RIGHT
360     IF I$ = "D" THEN GOTO 600 : REM DELETE ALLOPHONE
370     IF I$ = "I" THEN GOTO 700 : REM TURN ON INSERT MODE
372     IF LEN I$ = 1 THEN GOTO 900
380     GOTO 800
400     REM MOVE LEFT
410     IF P<2 THEN GOTO 900
420     LET P=P-1
430     GOTO 300
500     REM MOVE RIGHT
510     IF (P>L) OR (P>LENGTH - 1) THEN GOTO 900
530     LET P=P+1
540     GOTO 300
600     REM DELETE ALLOPHONE
610     IF P>L THEN GOTO 300
620     FOR I=P TO L
630     LET S(I) = S(I+1)
640     NEXT I
650     LET L = L-1
660     GOTO 300
700     REM TURN ON INSERT MODE
710     LET M = M-1
715     IF M<0 THEN M=1
720     GOTO 300
800     REM TEST FOR VALID ALLOPHONE ENTRY
810     FOR I = 1 TO 64
820     LET X$ = I$
830     FOR J=1 TO 2
832     LET Q = CODE(X$)
834     LET W = Z((I-1)*3+J)
840     IF (Q<W) OR (Q>W) THEN GOTO 890
850     LET X$ = TL$(X$)
860     NEXT J
870     IF X$ = ""THEN GOTO 950
880     IF CODE(X$) = Z((I-1)*3+3) THEN GOTO 950
890     NEXT I
900     REM USER ENTERED INVALID DATA
910     PRINT "***INVALID ENTRY***"
920     GOTO 315 : REM RETURN TO MAINLINE, DON'T
        REFRESH SCREEN
950     REM USER ENTERED VALID ALLOPHONE
960     IF P>LENGTH THEN GOTO 900
```

```
 970      IF M=0 THEN GOTO 1100
 980      IF L>LENGTH -1 THEN GOTO 900 : REM ERROR-NO
          ROOM TO INSERT
 990      LET X = S(P)
1000      FOR J = P+1 TO L+1
1010      LET Y = S(J)
1020      LET S(J) = X
1030      LET X=Y
1040      NEXT J
1050      LET L = L+1
1100      REM NOW PUT ALLOPHONE INTO STRING AT POS P
1110      LET S(P) = 1
1120      IF L<P THEN LET L = P
1130      LET P = P+1
1150      GOTO 300
2000      REM DISPLAY ALLOPHONE STRING
2010      CLS
2030      IF P=1 THEN PRINT "->";
2035      IF L=0 THEN GOTO 2090
2040      FOR I = 1 TO L
2060      FOR J = 1 TO 3
2062      PRINT CHR$(Z((S(I)-1)*3+J));
2064      NEXT J
2066      PRINT "b";
2070      IF I=P-1 THEN PRINT "->";
2080      NEXT I
2090      PRINT
2105      IF M=1 THEN PRINT "***INSERT MODE***"
2110      RETURN
3000      REM SECTION TO MAKE BOARD PRONOUNCE PHRASE
3020      FOR I=1 TO L
3050      POKE(B+8), S(I)-1 : REM/POKE ALLOPHONE INTO
          MACHINE CODE
3060      LET R=USR(B) : REM/PRONOUNCE ALLOPHONE
3070      NEXT I
3080      POKE(B+B), 0
3090      LET R=USR(B)
3100      GOTO 300
4000      STOP
```

TEST PROGRAM

The following program allows you to build words or phrases using the decimal codes of the allophones (see Table 9-1) rather than the allophone symbols used in the Exclusive Phrase Finder Program. This program is advantageous in that it can operate without the use of an external memory device. However, it is not an efficient program and should only be used when testing the synthesizer, hence its name.

For the ZX80, replace line 70 with POKE 16435,A(I), and replace line 80 with LET L = USR(16427). The question mark in line 15 must be replaced by the number of allophones to be entered.

```
15    LET LENGTH=?
20    DIM A(LENGTH)
30    FOR I=1 TO LENGTH
40    INPUT A(I)
50    NEXT I
60    FOR I=1 TO LENGTH
70    POKE 16522,A(I)
80    LET L = USR(16514)
90    NEXT I
```

Test Program Description

The above program must be entered *after* the Machine Code Program is entered. While entering the Test Program, note line 15 LET LENGTH=? This statement tells the computer the number of allophones that are needed to speak the word or phrase you have chosen. The question mark must be replaced with the number of allophones that will be used. For example:

LET LENGTH=25	The word to be spoken is made up of 25 allophones.

After typing in the Test Program and running it, the allophone codes for your desired word or phrase must be entered.

After the last allophone code is entered, the computer will automatically speak the word or phrase. Therefore, the last allophone should always be a pause. This will cause the speech synthesizer to stop talking once the word or phrase has been completed. To repeat the word or phrase, type

GOTO 60 "ENTER"

After the phrase is spoken, you must RUN the program again to enter a new phrase. You must always enter the exact number of allophones as specified in line 15. If your word or phrase is shorter than the length specified in line 15, just pad the last few allophones with pauses. If your word is longer than the length

specified, the computer will only speak the number of allophones originally specified.

ALLOPHONES TO DECIMAL CODES

If you would like to make your computer say "HELLO," the following steps are required:

1. You must first specify in line 15 how many allophones make up the word "hello." As explained in Chapter Three (see Table 3-1) the allophones required to say "hello" are HH1 EH LL OW PA3. That's a total of five allophones. At this point (cf. Table 9-1) note the decimal codes that are associated with each allophone. For example:

Allophones	HH1	EH	LL	OW	PA3
Decimal Codes	27	7	45	53	2

2. Since hello is made up of five allophones, enter the number 5 in line 15.
3. RUN the program. The screen will be cleared.
4. One at a time, the decimal codes associated with the word hello:

```
27   "ENTER"
 7   "ENTER"
45   "ENTER"
53   "ENTER"
 2   "ENTER"
```

Once the last decimal code has been entered, the speech synthesizer will say *hello.*

SAMPLE PROGRAM

The following program describes how to add "N" phrases to your existing programs. First the Machine Code Program must be en-

tered and stored in a REM statement; then your program can be entered using the following statements:

Data Statements

These statements must appear in the program before the lines that enable the synthesizer to speak.

> NOTE: The decimal codes for each particular allophone (as shown in Table 9-1) must be inserted in the places denoted by an asterisk.

```
100    LET S$(1)=CHR$(*)+ . . . . +CHR$(64)
110    LET S$(2)=CHR$(*)+ . . . . +CHR$(64)
120    LET S$(3)=CHR$(*)+ . . . . +CHR$(64)
130    LET S$(N)=CHR$(*)+ . . . . +CHR$(64)
```

Subroutine

```
10000    LET N=1
10010    LET A=CODE(S$(X,N))
10020    POKE(B+8),A
10030    LET R=USR(B)
10040    IF A=64 THEN RETURN
10050    LET N=N+1
10060    GOTO 10010
```

First Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the first phrase.

```
1000    LET X=1
1010    GOSUB 10000
```

Second Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the second phrase.

```
2000    LET X=2
2010    GOSUB 10000
```

Third Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the third phrase.

```
3000    LET X=3
3010    GOSUB 10000
```

Nth Phrase to Be Spoken. These lines must appear in the program directly after the line you want the synthesizer to speak the Nth phrase.

```
N000    LET X=N
N010    GOSUB 10000
```

LET'S GET TECHNICAL . . .

DRIVER SUBROUTINES

This next section describes the driver, or talk subroutines, for the T/S 1000, ZX80, and ZX81. These driver subroutines can be used directly in your own application programs, or studied as examples.

```
  10    REM (Machine Code Program)
3020    FOR I=1 TO L
3050    POKE(B+8), S(I)−1
3060    LET R=USR(B)
3070    NEXT I
```

Explanation of the Machine Code Program

*MEMORY LOCATION**	*MACHINE CODE*	*OPERATION*	*COMMENTS*
16514	219 3	IN A,3	The computer reads the data on the 8 address lines of the speech chip from port 3.

*MEMORY LOCATION**	*MACHINE CODE*	*OPERATION*	*COMMENTS*
16516	230 1	AND A,1	This tests the state of $\overline{\text{LRQ}}$ (busy line) from D0.
16518	202 130† 64	JPZ (16514)	If $\overline{\text{LRQ}}$ is low, the chip is not ready and will jump back to 16514.
16521	62 0	LD A, "Allophone"	Puts allophone address from basic program into "A."
16523	211 3	OUT A,3	Sends allophone address to speech chip.
15525	201	RET	Returns to Basic program.

* For the ZX80 the machine code locations are 16427 to 16438.
† For the ZX80 the machine code address 130 should be 43.

Explanation of the Basic Program

Line 3020 FOR I=1 TO L

L represents the number of allophones to be spoken which are stored in array S.

Line 3050 POKE(B+8), S(I)−1

Each allophone was stored in memory at its decimal location plus 1. This statement subtracts 1 from the allophone code and sends it to the Machine Code Program at location B+8 or 16514+8 or 16522.

Line 3060 LET R=USR(B)

This statement calls the user subroutine 16514 through 16525.

Line 3070 NEXT I

The next allophone to be spoken.

DEBUGGING YOUR CIRCUIT

The following steps are simple test procedures to follow when debugging your speech synthesis circuit. Please note that additional test equipment may be required (e.g., oscilloscope, voltmeter, logic probe). In addition to the test equipment, you should be experienced with debugging electrical circuits.

Procedures to Ensure That Your Connections Are Correct

1. Check all the power and ground connections. Pins 7, 19, and 23 of the SPO256-AL2 are +5 V. Pins 2 and 22 of the SPO256-AL2 are negatively grounded.
2. Disconnect the audio filter circuit from pin 24. Turn up the volume on the LM386 to maximum. Send another audio signal through the audio filter circuit. The output of a transistor radio is sufficient. Tune into a station where speech is playing rather than music. If the signal is heard at the speaker, this circuit is OK (it may be distorted somewhat). When resetting the chip, pin 24 should go to a logic 0 level. When the device is not speaking, a 40-kHz square wave is present on this pin.
3. Pins 2 and 25 of the SPO256-AL2 should be a logic 1 (+5 V). Pressing the reset button should change it to a logic 0 (0 V) and a small click should be heard in the speaker. If this doesn't occur and Procedures 1 and 2 are OK, the problem has to be in the 100-kΩ resistor, the push button switch, or the SPO256-AL2.
4. The signal on pin 26 should be oscillating. If this does not occur, either the crystal is bad or the capacitors connected to ground are the wrong capacitance. An oscilloscope or logic probe is required for this test.
5. Disconnect all the address lines (pins 10, 11, 13–18) from

your microcomputer and then ground the pins. Connect pin 15 to +5 V. Disconnect the $\overline{\text{ALD}}$ (pin 20) and connect a momentary switch between it and +5 v. Upon depressing the momentary switch, allophone KK3 should be heard. If Steps 1–4 are operating properly and this step fails, the SPO256-AL2 is malfunctioning.

6. If Steps 1–5 are operating, proceed with the following: For the T/S 1000 and ZX81, enter its' Machine Code Program. Then enter POKE 165,225 and LET L=USR(16514). For the ZX80, enter its Machine Code Program. Then enter POKE 16435,5 and LET L=USR(16427). Immediately on entering LET L=USR(16514) or LET L=USR(16427), the state of the $\overline{\text{LRQ}}$ line (pin 9 of the SPO256-AL2) will change from a logic 0 to a logic 1. If this does not occur, then carefully recheck all your wiring and try substituting new TTL logic parts (i.e., 74LS27, 74LS368).

NOTE: An oscilloscope is required to debug the logic parts. Since these parts are so inexpensive, I suggest you replace them before testing them.

CHANGING THE PORT LOCATION

Lines A4, A5, and A7 are decoded with a NOR gate (74LS27). To change the port locations, either one of these port locations must be disconnected and another address line must be connected in its place. For example:

A0	A1	A2	A3	A4	A5	A6	A7
X	X	X	X	0	0	X	0

The synthesizer will speak when A4, A5, and A7 are 0. All possible ports with A4, A5, and A7 = 0 are 0–15 and 64–79. (X is a "don't care" condition). To change the port location, jump another address line where A4, A5, or A7 was previously connected. For

For example: Step 1—Disconnect A4. Step 2—Connect A6 where A4 was connected.

A0	A1	A2	A3	A4	A5	A6	A7
X	X	X	X	X	0	0	0

Now ports 0–31 are being used. Since only three address lines are being decoded, a total of 32 port locations will always be occupied.

CHAPTER NINE

TECHNICAL INFORMATION ABOUT YOUR SPEECH SYNTHESIZER CHIP

This chapter describes the hardware requirements for the speech synthesizer chip chosen. The schematics included are not designed for a specific computer. The intent is to familiarize the reader with the general operation of a speech synthesizer. The information that follows is for the *experienced* machine language programmer or designer who wants to interface this device to an 8-bit microprocessor or microcomputer. In addition to being an experienced electronics designer, one must be versed in transistor transistor logic (TTL), machine language programming, and be able to read technical information from a data sheet. Additional hardware equipment may also be required to test the circuit.

The General Instrument Allophone Speech Synthesizer (SPO256-AL2) will be used as an example. Its simple interface provides a complete speech package on a single chip.

GENERAL DESCRIPTION

This speech processor (described in General Instrument, SPO256 Data Sheet, 1983) is a single chip N-channel (negatively doped conducting channel-type transistor), MOS (metallic oxide semiconductor), LSI (large-scale integration) device that can be programmed to synthesize speech or sound effects. The achievable output is equivalent to a flat frequency response ranging from 0 to 5 kHz. It incorporates four basic functions:

1. A software programmable digital filter that can be made to model the human vocal tract. Its construction is a 6-stage, 12-pole cascade digital filter as described in the Let's Get Technical section of Chapter Two.
2. A 16K ROM which stores both the allophone data and the program instructions for the microcontroller. Table 9-1 con-

Table 9-1 ALLOPHONE ADDRESSES*

DECIMAL CODES	ALLOPHONE	SAMPLE WORD	DURATION (ms)
0	PA1	*(pause)*	10
1	PA2	*(pause)*	30
2	PA3	*(pause)*	50
3	PA4	*(pause)*	100
4	PA5	*(pause)*	200
5	OY	Boy	420
6	AY	Sky	260
7	EH	End	70
8	KK3	Comb	120
9	PP	Pow	210
10	JH	Dodge	140
11	NN1	Thin	140
12	IH	Sit	70
13	TT2	To	140
14	RR1	Rural	170
15	AX	Succeed	70
16	MM	Milk	180
17	TT1	Part	100
18	DH1	They	290
19	IY	See	250
20	EY	Beige	280
21	DD1	Could	70
22	UW1	To	100
23	AO	Aught	100
24	AA	Hot	100
25	YY2	Yes	180
26	AE	Hat	120
27	HH1	He	130

DECIMAL CODES	ALLOPHONE	SAMPLE WORD	DURATION (ms)
28	BB1	Business	80
29	TH	Thin	180
30	UH	Book	100
31	UW2	Food	260
32	AW	Out	370
33	DD2	Do	160
34	GG3	Wig	140
35	VV	Vest	190
36	GG1	Guest	80
37	SH	Ship	160
38	ZH	Azure	190
39	RR2	Brain	120
40	FF	Food	150
41	KK2	Sky	190
42	KK1	Can't	160
43	ZZ	Zoo	210
44	NG	Anchor	220
45	LL	Lake	110
46	WW	Wool	180
47	XR	Repair	360
48	WH	Whig	200
49	YY1	Yes	130
50	CH	Church	190
51	ER1	Fir	160
52	ER2	Fir	300
53	OW	Beau	240
54	DH2	They	240
55	SS	Vest	90
56	NN2	No	190
57	HH2	Hoe	180
58	OR	Store	330
59	AR	Alarm	290
60	YR	Clear	350
61	GG2	Got	40
62	EL	Saddle	190
63	BB2	Business	50

* From General Instrument, Allophone Application Report, 1982, reprinted with permission from General Instrument.

tains the decimal address locations where the allophones are stored in memory.

3. A microcontroller which serves several purposes. It reads the data pins of the SPO256-AL2 to determine the starting address of each new sound sequence. Once determined, it goes to the address in the ROM where the data to synthesize the sound is located. It then supplies the data to the digital filter. These are the 12 parameters of data as described in Chapter Two. It also finds and delivers the necessary amplitude, pitch, and duration information to the filter.
4. A pulse width modulator that creates a digital output that is converted to an analog signal when applied to a low pass filter with a nominal cut-off frequency of 5 kHz. The signal is then amplified to drive any 4- or 8-Ω speaker.

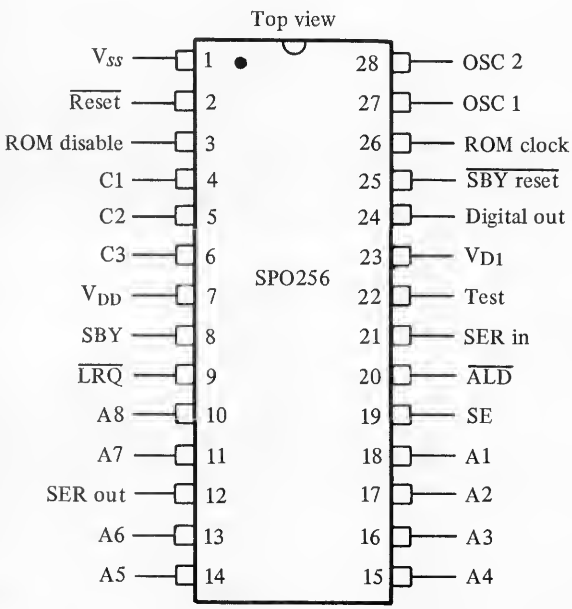


Fig. 9-1 Pin configuration of the SPO256 (28 lead dual in line). This diagram was reprinted from the General Instrument SPO256 Data Sheet, 1983, with permission from General Instrument.

Since this system incorporates all the facets of a speech system (i.e., voice generator, memory, controller, and digital-to-analog converter), you need not have the knowledge of the internal operation of the synthesizer chip. Simply output a decimal code to the chip to cause the desired utterance to be produced.

Figure 9-1 shows the pin configuration of the SPO256. Table

Table 9-2 PIN FUNCTIONS

PIN NUMBER	NAME	FUNCTION
1	V_{ss}	Ground
2	$\overline{\text{Reset}}$	A logic† 0 resets the speech ship; must be returned to a logic 1 for normal operation.
3	ROM disable	For use with an external serial speech ROM. A logic 1 disables the external ROM.
4,5,6	C1,C2,C3	Output control lines used by an external serial speech ROM.
7	V_{DD}	Primary power supply.
8	SBY	*Standby.* A logic 1 output indicates that the speech chip is inactive (not talking) and V_{DD} can be powered down externally to conserve power. When the speech chip is reactivated, by an address being loaded, SBY will go to a logic 0.
9	$\overline{\text{LRQ}}$	*Load request.* $\overline{\text{LRQ}}$ is a logic 1 output whenever the input buffer is full. When $\overline{\text{LRQ}}$ goes to logic 0, the input port is loaded by placing the 8 address bits on A1–A8 and pulsing the $\overline{\text{ALD}}$ input (see pin 20).
10,11,13,14, 15,16,17,18	A8,A7,A6,A5, A4,A3,A2,A1	Eight-bit address which defines any one of 256 speech entry points. For the SPO256-AL2, only six bits are used.

Table 9-2 (*Continued*)

PIN NUMBER	NAME	FUNCTION
12	SER out	*Serial address out.* This output transfers a 16-bit address serially to an external speech ROM.
19	SE	*Strobe enable.* Normally held in a logic 1 state.
20	$\overline{\text{ALD}}$	*Address load.* A negative pulse on this input loads the 8 address bits into the input port. The leading edge of this pulse causes $\overline{\text{LRQ}}$ to go high.
21	SER in	*Serial in.* This is an 8-bit serial data input from an external speech ROM.
22	TEST	A logic 1 places the SPO-256 in the test mode. This pin should normally be grounded.
23	V_{D1}	Standby power supply for the interface logic and controller.
24	Digital out	Pulse width modulated digital speech output which, when filtered by a 5-kHz low-pass filter and amplified, will drive a loudspeaker.
25	$\overline{\text{SBY reset}}$	*Standby reset.* A logic 0 resets the interface logic; it normally should be a logic 1.
26	ROM clock	1.56-MHz clock for an external serial speech ROM.
27	OSC 1	*XTAL in.* Input connection for a 3.12-MHz crystal.
28	OSC 2	*XTAL out.* Output connection for a 3.12-MHz crystal.

* General Instrument, SPO256 Data Sheet, 1983, reprinted with permission from General Instrument.

† All logic levels are TTL compatible.

9-2 explains each pin function. You need not concern yourself with pins 3, 4, 5, 6, 8, 12, 21, and 26. These pins are used with custom ROMs supplied by General Instrument for memory expansion.

Table 9-3 explains the dc and ac characteristics of the device. Figure 9-2 shows a typical timing diagram for each data input.

For a standard 8-bit microprocessor or microcomputer you need to be familiar with the requirements in Table 9-4.

Table 9-3 ELECTRICAL CHARACTERISTICS*†

CHARACTERISTICS	SYMBOL	MINIMUM	MAXIMUM	UNITS
Direct Current‡				
Primary supply voltage	V_{DD}	4.6	7	V
Standby supply voltage	V_{D1}	4.6	7	V
Primary supply current	—	—	90	mA
Standby supply current	—	—	—	mA
Inputs				
A1–A8, $\overline{\text{ALD}}$, SER in, TEST, SE				
Logic 0	—	0	0.6	V
Logic 1	—	2.4	V_{D1}	V
Capacitance	—	—	10	pF
Leakage	—	—	10	μA
Reset, $\overline{\text{SBY Reset}}$				
Logic 0	—	0	0.6	V
Logic 1	—	3.6	V_{D1}	V
Oscillator leakage OSC 1	—	1.0	10	μA
Outputs				
SBY, Digital out, C1, C2, C3, $\overline{\text{LRQ}}$, ROM DISABLE, ROM CLOCK, SER out				
Logic 0	—	0	0.6	V
Logic 1	—	3.5	V_{D1}	V

Table 9-3 (*Continued*)

CHARACTERISTICS	SYMBOL	MINIMUM	MAXIMUM	UNITS
Alternating Current‡				
Reset, SBY Reset	t(pw1)	100	—	μs
$\overline{\text{ALD}}$	t(pw2)	200	800	ns
A1–A8 Set Up	t(s)	160	—	ns
A1–A8 Hold	t(h)	160	—	ns
$\overline{\text{LRQ}}$	t(o)	—	640	ns

* This diagram was reprinted from the General Instrument SPO256 Data Sheet, 1983, with permission from General Instrument.
† Maximum ratings: V_{D1} and V_{DD} measure −.03 to −12 V; storage temperature measures −25 to +125°C. The clock crystal frequency is 3.12 MHz.
‡ Operating temperature T(A) is 0–70°C.

OPERATION

The operation of the speech synthesizer is fairly simple. After wiring up the clock circuit, reset circuit, audio filter circuit, power, and ground connections (see Chapters Four, Five, Six, Seven, or Eight), your 8-bit microcontroller need only supply the address data and the address load ($\overline{\text{ALD}}$) pulse for the appropriate sound to be spoken. Upon receiving the address load pulse, the synthesizer

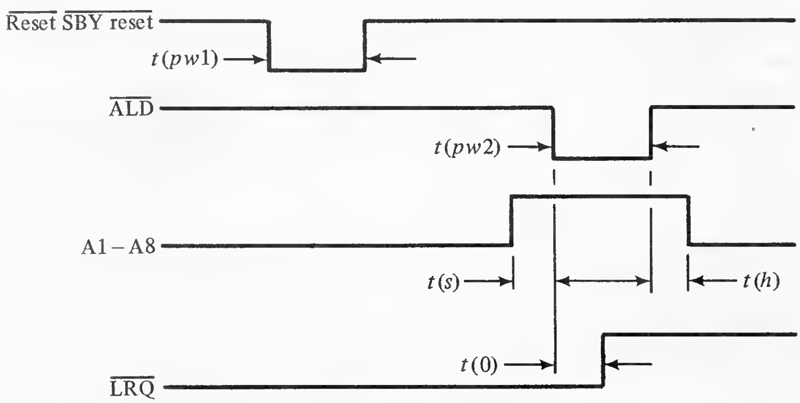


Fig. 9-2 Timing diagram of the SPO256.

will perform two functions. It generates a load request ($\overline{\text{LRQ}}$) pulse (*high*) back to your microcontroller. This signals the microcontroller to wait until the synthesizer has finished speaking before it can accept the next address. The synthesizer's other function is to process the address data that is received in order to speak the appropriate sound. Once the sound is completed, the synthesizer

Table 9-4 MICROPROCESSOR ADDRESS REQUIREMENTS*

DESCRIPTION	PIN NUMBERS	COMMENTS
Power	7, 19, and 23	+5 V
	2 and 25	Connected to +5 V via a 100-kΩ resistor
	1 and 22	Negatively grounded
Digital out	24	Connects to an audio filter amplifier and then to a speaker.
Address lines A8–A1	10, 11, 13, 14, 15, 16, 17, and 18	These lines allow an address to be loaded into the chip.
$\overline{\text{Address load}}$ ($\overline{\text{ALD}}$)	20	A negative pulse on this input loads the 8 address bits into the input port. This pulse causes the load request to go high.
$\overline{\text{Load request}}$ ($\overline{\text{LRQ}}$)	9	This line is a logic 1 whenever the device is active (talking). When $\overline{\text{LRQ}}$ goes to a logic 0, the next address can be loaded.

Table 9-4 (*Continued*)

DESCRIPTION	PIN NUMBERS	COMMENTS
Timing control†	27 and 28	The timing is supplied via a 3.12-MHz clock connected to pins 27 and 28 of the speech synthesizer.

* From General Instrument, SPO256 Data Sheet, 1983, reprinted with permission from General Instrument.

† The capacitors connected to the ground shown in Fig. 9-3 are 22 pF. These are used when a 3.12-MHz *crystal oscillator* is used. The diagrams in Chapters Four, Five, Six, Seven, and Eight show 100-pF capacitors connected to the ground. These are used when a 3.12-MHz *ceramic resonator* is used. (The ceramic resonator supplied with all the kits is a MURATA CSA 3.12-MS2 ceramic resonator.)

sends another load request pulse (*low*) back to your microcontroller. This signals the microcontroller that the synthesizer is now ready to accept the next address. This process, when performed repeatedly in the program of your microcontroller, can generate complete words, phrases, and even sentences. The choice of the words is dependent on the code you've written for your microcontroller. The addresses you use to create your words should correspond to the addresses listed in Table 3-1 (see Chapter Three for an explanation regarding creating words from allophones). The only difference is that your microcontroller is supplying a binary (base 2) representation of the decimal (base 10) codes to pins A1–A6 of the SPO256-AL2.

The Timing is Critical

The timing requirements for the address lines and the address load pulse are critical. If the address data is not present on the lines at the time of the address load pulse, the speech synthesizer will not speak. The synthesizer may even "crash" (static noise emitting out of the speaker). This will also happen in attempting to address

the synthesizer beyond the 0–63 decimal boundaries. To stop the noise for a manual reset, just depress the reset button. For a computer controlled reset, the minimum pulse is 100 ms (see Fig. 9-2 and Table 9-3).

The address lines (A1–A8) require a set up and hold time of 160 ns. The address load pulse ($\overline{\text{ALD}}$) must be between 200–800 ns. This means, if we assume an address load pulse of 500 ns, the address data must be present on the address lines for a minimum of 820 ns. It must be present for 160 ns before (set up) the $\overline{\text{ALD}}$ pulse and 160 ns after (hold) the $\overline{\text{ALD}}$ pulse. These pulses will cause the first address to be spoken. When $\overline{\text{LRQ}}$ goes low, as stated before, the next set of pulses for the next address can be generated.

Figure 9-3 shows an application of the SPO256-AL2 with any 8-bit microprocessor or microcomputer. The SPO256-AL2 operates as a complete speech system and will speak any sound sequences entered into it via the microcontroller. As already explained, the microcontroller need only specify the 6-bit codes of the sounds to be synthesized. This represents a very minute load on the microcomputer or microprocessor. This means that your microcomputer can use the synthesizer as an output device for many programs without significantly slowing down or interfering with their operation.

TESTING YOUR CIRCUIT

Procedures to Ensure That Your Connections Are Correct

1. Check all the power and ground connections. Pins 7, 19, and 23 of the SPO256-AL2 are +5 V. Pins 2 and 22 of the SPO256-AL2 are negatively grounded.
2. Disconnect the audio filter circuit from pin 24. Turn up the volume on the LM386 to maximum. Send another audio signal through the audio filter circuit. The output of a transistor radio is sufficient. Tune into a station where speech is playing rather than music. If the signal is heard on the

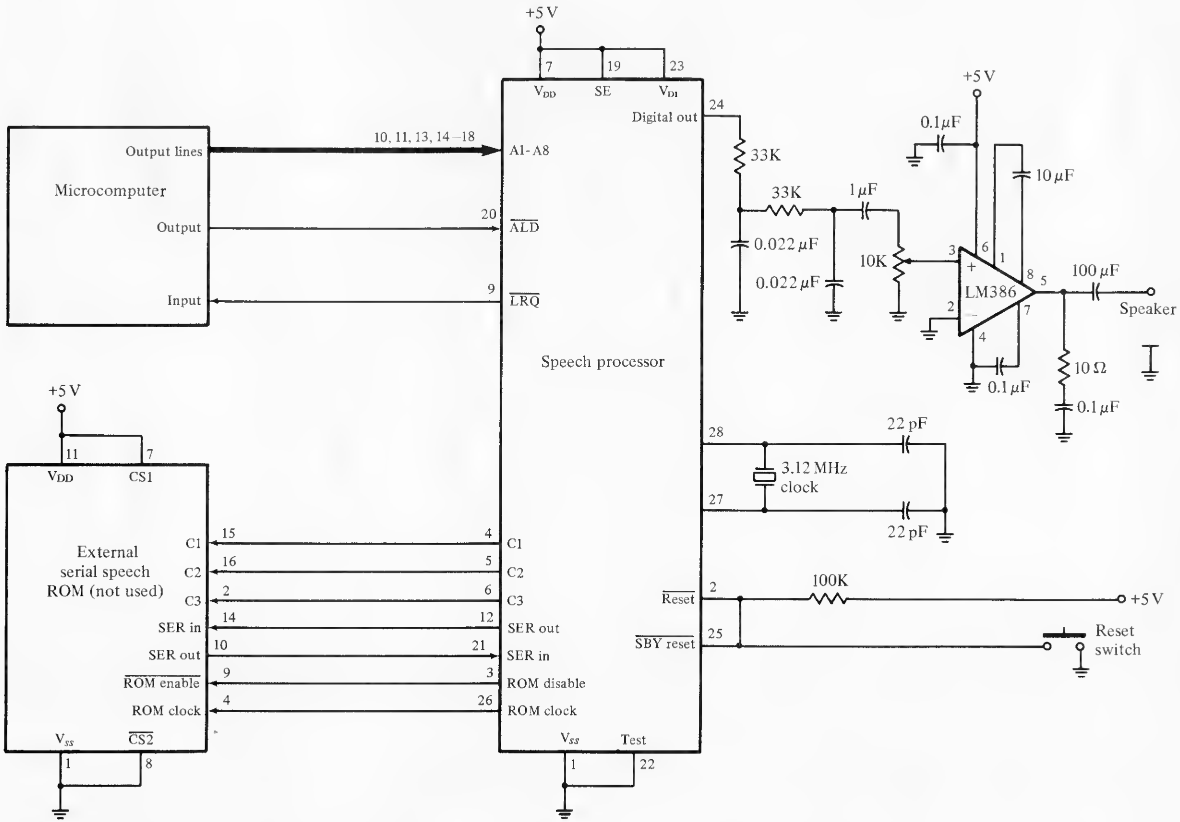


Fig. 9-3 Typical application of the SPO256 interfaced with a standard 8-bit microcontroller. The external serial speech ROM is optional and not used with the SPO256-AL2. This schematic was reprinted from the General Instrument SPO256 Data Sheet, 1983, with permission from General Instrument.

speaker, this circuit is OK (it may be distorted somewhat). When resetting the chip, pin 24 should go to a logic 0 level. When the device is not speaking, a 40-kHz square wave is present on this pin.

3. Pins 2 and 25 of the SPO256-AL2 should be a logic 1 (+5 V). Pressing the reset button should change it to a logic 0 (0 V) and a small click should be heard in the speaker. If this doesn't occur and procedures 1 and 2 are OK, the problem has to be in the 100-kΩ resistor, the push button switch, or the SPO256-AL2.
4. The signal on pin 26 should be oscillating. If this does not occur, either the crystal is bad or the capacitors connected to ground are the wrong capacitance. An oscilloscope or logic probe is required for this test.
5. Disconnect all the address lines (pins 10, 11, 13–18) from your microcomputer and then ground the pins (except pin 15). Connect pin 15 to +5 V. Disconnect the $\overline{\text{ALD}}$ (pin 20) and connect a momentary switch between it and +5 V. Upon depressing the momentary switch, allophone KK3 should be heard. If Steps 1–4 are operating properly and this step fails, the SPO256-AL2 is malfunctioning.
6. If all of the above items are operating, then carefully recheck all your wiring and try substituting parts that are known to be functioning properly (i.e., TTL chips, latches, etc.), where possible, for the same parts in the circuit.

CHAPTER TEN

INTRODUCTION TO TEXT-TO-SPEECH CONVERSION

This book has explored the area of adapting speech synthesis to your home computer through a technique called allophone synthesis. We discussed how speech is made up of small "subunits," and how words and phrases can be created by joining these subunits together. But you must make the decision as to which allophones are to be used in the programs when creating a desired phrase.

WHAT IS TEXT-TO-SPEECH?

Your programs can be made to speak any word; for example, if you want to speak the word "hello," you choose allophones from the list provided in Chapter Three. However, you must continue to do this for every word that you want your program to speak. Wouldn't it be nice if you could simply type in the desired word, the way it is spelled, and have the computer assemble the allophone automatically? Such a process is commonly referred to as *text-to-speech conversion*. Using our example above, to get your computer to speak the word "hello," simply type **hello** and press RETURN or ENTER (depending on the computer). The computer would then sound out the word using its own internal program. Some systems may also present you with a list of allophones necessary to say that word.

With this type of system, the computer can actually read aloud messages typed into the computer keyboard or read messages displayed on the computer screen. As you can see, this system offers many advantages. It is much easier and much more convenient to use, and it is especially useful when developing vocabularies for your own programs. It simplifies the laborious task of creating words and phrases from allophones, and makes the speech synthesizer a lot more enjoyable and fun to use.

LETTER-TO-SOUND RULE SET

The text-to-speech conversion itself is accomplished by what is known as a *letter-to-sound rule set.* The letter-to-sound rule set is a simple list of instructions which tells what allophone to use for a given letter, depending on its surrounding letters. Let's use the word "pale" as an example. First we type **p a l e** into the computer and press RETURN. The computer now has to determine how that word sounds or is spoken. The program takes the first letter **p** and looks into its set of statements describing how to use allophones for the letter **p.** As it turns out, there is only one allophone that is ever used for **p**—the PP allophone. The next letter, **a,** is a little more complicated. The **a** may be a long **a** as in "base" or a short **a** as in "pat." There is a rule that says, when **a** is followed by a consonant which is followed by the vowel **e** ending the word, the allophone which represents the long **a** sound is EY. This is the simple rule that we all learned in grade school for sounding out words.

Let's try our original example and see how the letter-to-sound program pronounces the word "hello." First it goes to the **h** rules. If the **h** is followed by an **e,** and then a consonant, and the word does not end in **e,** the HH1 allophone is used. It then goes to the **e** rules—all the rules are special cases when using **er** or **ew.** Since these rules don't apply, and the word does not end with an **e,** the EH allophone is used. Next, it goes to the **l** rules. Since there is only one allophone to pronounce an **l,** the LL allophone is used. Then it goes to the l rules again. There is a special

rule that says if an **l** has another **l** before it, the second **l** is silent. Finally it goes to the **o** rules. An **o** followed by a space is pronounced with the OW allophone. So the final allophone code that the letter-to-sound rules chose is HH1 EH LL OW.

But a system of this type might contain several hundred such rules. Furthermore, the English language has many exceptions to its own pronunciation rules, as any of us know if we try to sound out a word we have never seen before, and this type of text-to-speech system alone will not get the words correct 100 percent of the time. The rule set itself can be expanded to include many special cases, but even then there are always exceptions.

ADDING A DICTIONARY

The system can be further augmented by the addition of a lexicon (dictionary). These are really specialized rules which only apply to given groups of letters, usually a specific word. For example, we might use a rule which consists of "when you find the letters **b u i l d**, you will say the allophones BB2 IH LL PA2 DD1," which then results in the correct pronunciation of the word *"build."* This is an augmentation to text-to-speech since it helps cover cases which true text-to-speech (letter-to-sound rule set) simply will not handle. It is interesting to note that even if we formed all the words in the English language as specific rules, we still would not get the correct pronunciation 100 percent of the time. An example of a failure would be the word "read." In order to determine whether it should be pronounced "red" or "reed," the system must understand the letters involved as well as the words that surround the word "read." This becomes an artificial intelligence problem and involves an entirely new level of complexity.

Although text-to-speech becomes very accurate with the use of an extensive dictionary, the memory and expense to store it would be phenomenal. But minimum dictionary storage of certain words that are hard to pronounce can increase the accuracy of the text-to-speech system for minimal additional cost and can, therefore, be a very useful tool.

INTONATIONS

The addition of emotional content and intonation to machine-generated speech also involves artificial intelligence or true understanding of what is being said on the part of the machine. If a text-to-speech system randomly selects stressed syllables, the speech tends to sound very unnatural and unintelligible. Both stress and pitch must be precisely applied to be effective in a text-to-speech system. For this reason, text-to-speech conversion systems typically use a "robot-like" voice with little or no emotion.

Because the English language is so complex, more and more speech rules are required to increase the accuracy of the text-to-speech system. This in turn increases the cost of the system. Therefore, a reasonable solution to the problem can be achieved by using a combination of the above rules. Of course, the optimum system attainable would sound like a human being and be 100 percent accurate. However, a text-to-speech system that comes close to this level of accuracy and quality would operate on a mainframe computer and cost approximately $100,000. This figure is an unrealistic figure for any small business to pay, never mind the home computer user! Reducing the cost is possible, and some text-to-speech systems for personal computers cost as little as $30. However, with speech synthesis, there is always a trade-off. To reduce the cost, one must reduce the amount of memory required. Reducing the memory storage reduces the complexity and size of the software algorithm. Reducing the features of the software algorithm leads to an affordable text-to-speech system with "acceptable" speech quality and accuracy.

With all these factors in mind, the choice of what rule sets to use and if pitch should be applied or not is left up to the manufacturer of the text-to-speech system. It is his or her job to choose the rule set that occupies the least amount of memory, is cost effective, and offers the maximum accuracy and flexibility. In this way, the consumer can purchase a quality text-to-speech system at an affordable price.

A NOTE FOR ADVANCED READERS

A text-to-speech system can be developed utilizing the speech synthesis circuits explained in Chapters Four, Five, Six, Seven, and Eight (see availability in Appendix B). The programs may comprise letter-to-sound rules based on a set of rules developed by the Naval Research Laboratory (N.R.L.) in Washington, D.C. Most text-to-speech systems are based on the N.R.L. rules because of their high accuracy and their availability to the public. However, the development would require conversion of the basic N.R.L. rule set to the specific synthesizer and *advanced* programming capability.

SUMMARY

The addition of a text-to-speech algorithm to an allophone speech system carries with it many major advantages. The ability to read text entered via a keyboard and the ability to read aloud messages displayed on the computer screen are two of these advantages. A text-to-speech system also offers a more convenient approach in developing vocabularies. Although not 100 percent efficient, the addition of a lexicon to the text-to-speech algorithm offers acceptable accuracy at an affordable price. As with any hardware device, where software is written to maximize the device's capabilities, the text-to-speech system described in this chapter maximizes the capabilities of an allophone speech system. This system is an essential tool for advanced speech synthesis development.

APPENDIX A

ALLOPHONE DICTIONARY OF COMMONLY USED WORDS

The following list of words is provided to help you when building phrases for your programs. Reprinted from the General Instrument Allophone Application Note, 1983, with permission from General Instrument.) The allophone sounds corresponding to each word are the symbols that should be entered into the Exclusive Phrase Finder Programs listed in Chapters Four, Five, Six, Seven, and Eight. If you would like to enter speech into your own programs after testing it in the Exclusive Program, you may do so by converting the allophone sounds to decimal codes. After converting to the decimal codes, the codes should be stored in the Sample Program, also listed in Chapters Four, Five, Six, Seven, and Eight.

Keep in mind that the following allophone codes are suggestions only. As described in Chapter Three, what is pleasing to you may not satisfy someone else. So use this dictionary as a guideline only. If you want to use other sounds to pronounce the words, go right ahead!

LETTERS	*ALLOPHONES*
a	EY
b	BB2 IY
c	SS SS IY
d	DD2 IY
e	IY

LETTERS	*ALLOPHONES*
f	EH EH FF FF
g	JH IY
h	EY PA2 PA3 CH
i	AA AY
j	JH EH EY
k	KK1 EH EY
l	EH EH EL
m	EH EH MM
n	EH EH NN1
o	OW
p	PP IY
q	KK1 YY1 UW2
r	AR
s	EH EH SS SS
t	TT2 IY
u	YY1 UW2
v	VV IY
w	DD2 AX PA2 BB2 EL YY1 UW2
x	EH EH PA3 KK2 SS SS
y	WW AY
z	ZZ IY

NUMBERS	*ALLOPHONES*
zero	ZZ YR OW
one, won	WW SX AX NN1
two, to, too	TT2 UW2
three	TH RR1 IY
four, for, fore	FF FF OR
five	FF FF AY VV

NUMBERS	*ALLOPHONES*
six	SS SS IH IH PA3 KK2 SS
seven	SS SS EH EH VV IH NN1
eight, ate	EY PA3 TT2
nine	NN1 AA AY NN1
ten	TT2 EH EH NN1
eleven	IH LL EH EH VV IH NN1
twelve	TT2 WH EH EH LL VV
thirteen	TH ER1 PA2 PA3 TT2 IY NN1
fourteen	FF OR PA2 PA3 TT2 IY NN1
fifteen	FF IH FF PA2 PA3 TT2 IY NN1
sixteen	SS SS IH PA3 KK2 SS PA2 PA3 TT2 IY NN1
seventeen	SS SS EH VV EH NN1 PA2 PA3 TT2 IY NN1
eighteen	EY PA2 PA3 TT2 IY NN1
nineteen	NN1 AY NN1 PA2 PA3 TT2 IY NN1
twenty	TT2 WH EH EH NN1 PA2 PA3 TT2 IY
thirty	TH ER2 PA2 PA3 TT2 IY
forty	FF OR PA3 TT2 IY
fifty	FF FF IH FF FF PA2 PA3 TT2 IY
sixty	SS SS IH PA3 KK2 SS PA2 PA3 TT2 IY
seventy	SS SS EH VV IH NN1 PA2 PA3 TT2 IY
eighty	EY PA3 TT2 IY
ninety	NN1 AY NN1 PA3 TT2 IY
hundred	HH2 AX AX NN1 PA2 DD2 RR2 IH IH PA1 DD1

NUMBERS	*ALLOPHONES*
thousand	TH AA AW ZZ TH PA1 PA1 NN1 DD1
million	MM IH IH LL YY1 AX NN1

DAYS	*ALLOPHONES*
Sunday	SS SS AX AX NN1 PA2 DD2 EY
Monday	MM AX AX NN1 PA2 DD2 EY
Tuesday	TT2 UW2 ZZ PA2 DD2 EY
Wednesday	WW EH EH NN1 ZZ PA2 DD2 EY
Thursday	TH ER2 ZZ PA2 DD2 EY
Friday	FF RR2 AY PA2 DD2 EY
Saturday	SS SS AE PA3 TT2 PA2 DD2 EY

MONTHS	*ALLOPHONES*
January	JH AE AE NN1 YY2 XR 1Y
February	FF EH EH PA1 BR RR2 UW2 XR IY
March	MM AR PA3 CH
April	EY PA3 PP RR2 IH IH LL
May	MM EY
June	JH UW2 NN1
July	JH UW1 LL AY
August	AO AO PA2 GG2 AX SS PA3 TT1
September	SS SS EH PA3 PP PA3 TT2 EH EH MM PA1 BB2 ER1
October	AA PA2 KK2 PA3 TT2 OW PA1 BB2 ER1
November	NN2 OW VV EH EH MM PA1 BB2 ER1

MONTHS	*ALLOPHONES*
December	DD2 IY SS SS EH EH MM PA1 BB2 ER1

GENERAL	*ALLOPHONES*
alarm	AX LL AR MM
bathe	BB2 EY DH2
bather	BB2 EY DH2 ER1
bathing	BB2 EY DH2 IH NG
beer	BB2 YR
bread	BB1 RR2 EH EH PA1 DD1
by	BB2 AA AY
calendar	KK1 AE AE LL EH NN1 PA2 DD2 ER1
check	CH EH EH PA3 KK2
checked	CH EH EH PA3 KK2 PA2 TT2
checker	CH EH EH PA3 KK1 ER1
checkers	CH EH EH PA3 KK1 ER1 ZZ
checking	CH EH EH PA3 KK1 IH NG
checks	CH EH EH PA3 KK1 SS
clock	KK1 LL AA AA PA3 KK2
clown	KK1 LL AW NN1
cognitive	KK3 AA AA GG3 NN1 IH PA3 TT2 IH VV
collide	KK3 AX LL AY DD1
computer	KK1 AX MM PP1 YY1 UW1 TT2 ER
cookie	KK3 UH KK1 IY
coop	KK3 UW2 PA3 PP
correct	KK1 ER2 EH EH PA2 KK2 PA2 TT1

GENERAL	*ALLOPHONES*
corrected	KK1 ER2 EH EH PA2 KK2 PA2 TT2 PA2 DD1
correcting	KK1 ER2 EH EH PA2 KK2 PA2 TT2 IH NG
corrects	KK1 ER2 EH EH PA2 KK2 PA2 TT1 SS
crown	KK1 RR2 AW NN1
date	DD2 EY PA3 TT2
daughter	DD2 AO TT2 ER1
day	DD2 EH EY
divided	DD2 IH VV AY PA2 DD2 IH PA2 DD1
emotional	IY MM OW SH AX NN1 AX EL
engage	EH EH PA1 NN1 GG1 EY PA2 JH
engagement	EH EH PA1 NN1 GG1 EY PA2 JH MM EH EH NN1 PA2 PA3 TT2
engages	EH EH PA1 NN1 GG1 EY PA2 JH IH ZZ
engaging	EH EH PA1 NN1 GG1 EY PA2 JH IH NG
enrage	EH NN1 RR1 EY PA2 JH
enraged	EH NN1 RR1 EY PA2 JH PA2 DD1
enrages	EH NN1 RR1 EY PA2 JH IH ZZ
enraging	EH NN1 RR1 EY PA2 JH IH NG
equal	IY PA2 PA3 KK3 WH AX EL
equals	IY PA2 PA3 KK3 WH AX EL ZZ
error	EH XR OR
escape	EH SS SS PA3 KK1 PA2 PA3 PP

GENERAL	*ALLOPHONES*
escaped	EH SS SS PA3 KK1 PA2 PA3 PP PA2 TT2
escapes	EH SS SS PA3 KK1 PA2 PA3 PP SS
escaping	EH SS SS PA3 KK1 PA2 PA3 PP IH NG
extent	EH KK1 SS TT2 EH EH NN1 TT2
fir	FF ER2
freeze	FF FF RR1 IY ZZ
freezer	FF FF RR1 IY ZZ ER1
freezers	FF FF RR1 IY ZZ ER1 ZZ
freezing	FF FF RR1 IY ZZ IH NG
frozen	FF FF RR1 OW ZZ EH NN1
gauge	GG1 EY PA2 JH
gauged	GG1 EY PA2 JH PA2 DD1
gauges	GG1 EY PA2 JH IH ZZ
gauging	GG1 EY PA2 JH IH NG
hello	HH1 EH LL OW
hour	AW ER1
infinitive	IH NN1 FF FF IH IH NN1 IH PA2 PA3 TT2 IH VV
intrigue	IH NN1 PA3 TT2 RR2 IY PA1 GG3
intrigued	IH NN1 PA3 TT2 RR2 IY PA1 GG3 PA2 DD1
intrigues	IH NN1 PA3 TT2 RR2 IY PA1 GG3 ZZ
intriguing	IH NN1 PA3 TT2 RR2 IY PA1 GG3 IH NG
investigate	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH GG1 EY PA2 TT2

GENERAL	*ALLOPHONES*
investigated	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH GG1 EY PA2 TT2 IH PA2 DD1
investigates	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH GG1 EY PA2 TT1 SS
investigating	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH GG1 EY PA2 TT2 IH NG
investigator	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH GG1 EY PA2 TT2 ER1
investigators	IH IH NN1 VV EH EH SS PA2 PA3 TT2 IH GG1 EY PA2 TT2 ER1 ZZ
key	KK1 IY
legislate	LL EH EH PA2 JH JH SS SS LL EY PA2 PA3 TT2
legislated	LL EH EH PA2 JH JH SS SS LL EY PA2 PA3 TT2 IH DD1
legislates	LL EH EH PA2 JH JH SS SS LL EY PA2 PA3 TT1 SS
legislating	LL EH EH PA2 JH JH SS SS LL EY PA2 PA3 TT2 IH NG
legislature	LL EH EH PA2 JH JH SS SS LL EY PA2 PA3 CH ER1
letter	LL EH EH PA3 TT2 ER1
litter	LL IH IH PA3 TT2 ER1
little	LL IH IH PA3 TT2 EL
memories	MM EH EH MM ER2 IY ZZ
memory	MM EH EH MM ER2 IY
minute	MM IH NN1 IH PA3 TT2

GENERAL	*ALLOPHONES*
month	MM AX NN1 TH
nip	NN1 IH IH PA2 PA3 PP
nipped	NN1 IH IH PA2 PA3 PP PA3 TT2
nipping	NN1 IH IH PA2 PA3 PP IH NG
nips	NN1 IH IH PA2 PA3 PP SS
no	NN2 AX OW
physical	FF FF IH ZZ IH PA3 KK1 AX EL
pin	PP IH IH NN1
pinned	PP IH IH NN1 PA2 DD1
pinning	PP IH IH NN1 IH NG1
pins	PP IH IH NN1 ZZ
pledge	PP LL EH EH PA3 JH
pledged	PP LL EH EH PA3 JH PA2 DD1
pledges	PP LL EH EH PA3 JH IH ZZ
pledging	PP LL EH EH PA3 JH IH NG
plus	PP LL AX AX SS SS
ray	RR1 EH EY
rays	RR1 EH EY ZZ
ready	RR1 EH EH PA1 DD2 IY
red	RR1 EH EH PA1 DD1
robot	RR1 OW PA2 BB2 AA PA3 TT2
robots	RR1 OW PA2 BB2 AA PA3 TT1 SS
score	SS SS PA3 KK3 OR
second	SS SS EH PA3 KK1 IH NN1 PA2 DD1
sensitive	SS SS EH EH NN1 SS SS IH PA2 PA3 TT2 IH VV

GENERAL	*ALLOPHONES*
sensitivity	SS SS EH EH NN1 SS SS IH PA2 PA3 TT2 IH VV IH PA2 PA3 TT2 IY
sincere	SS SS IH IH NN1 SS SS YR
sincerely	SS SS IH IH NN1 SS SS YR LL IY
sincerity	SS SS IH IH NN1 SS SS EH EH RR1 IH PA2 PA3 TT2 IY
sister	SS SS IH IH SS PA3 TT2 ER1
speak	SS SS PA3 IY PA3 KK2
spell	SS SS PA3 PP EH EH EL
spelled	SS SS PA3 PP EH EH EL PA3 DD1
speller	SS SS PA3 PP EH EH EL ER2
spellers	SS SS PA3 PP EH EH EL ER2 ZZ
spelling	SS SS PA3 PP EH EH EL IH NG
spells	SS SS PA3 PP EH EH EL ZZ
start	SS SS PA3 TT2 AR PA3 TT2
started	SS SS PA3 TT2 AR PA3 TT2 IH PA1 DD2
starter	SS SS PA3 TT2 AR PA3 TT2 ER1
starting	SS SS PA3 TT2 AR PA3 TT2 IH NG
starts	SS SS PA3 TT2 AR PA3 TT1 SS
stop	SS SS PA3 TT1 AA AA PA3 PP
stopped	SS SS PA3 TT1 AA AA PA3 PP PA3 TT2
stopper	SS SS PA3 TT1 AA AA PA3 PP ER1
stopping	SS SS PA3 TT1 AA AA PA3 PP IH NG

GENERAL	*ALLOPHONES*
stops	SS SS PA3 TT1 AA AA PA3 PP SS
subject (*noun*)	SS SS AX AX PA2 BB1 PA2 JH EH PA3 KK2 PA3 TT2
subject (*verb*)	SS SS AX PA3 BB1 PA2 JH EH EH PA3 KK2 PA3 TT2
sweat	SS SS WW EH EH PA3 TT2
sweated	SS SS WW EH EH PA3 TT2 IH PA3 DD1
sweater	SS SS WW EH EH PA3 TT2 ER1
sweaters	SS SS WW EH EH PA3 TT2 ER1 ZZ
sweating	SS SS WW EH EH PA3 TT2 IH NG
sweats	SS SS WW EH EH PA3 TT2 SS
switch	SS SS WH IH IH PA3 CH
switched	SS SS WH IH IH PA3 CH PA3 TT2
switches	SS SS WH IH IH PA3 CH IH ZZ2
switching	SS SS WH IH IH PA3 CH IH NG2
system	SS SS IH IH SS SS PA3 TT2 EH MM
systems	SS SS IH IH SS SS PA3 TT2 EH MM ZZ
talk	TT2 AO AO PA2 KK2
talked	TT2 AO AO PA2 KK2 PA3 TT2
talker	TT2 AO AO PA2 KK1 ER1
talkers	TT2 AO AO PA2 KK1 ER1 ZZ
talking	TT2 AO AO PA2 KK1 IH NG
talks	TT2 AO AO PA2 KK2 SS

GENERAL	*ALLOPHONES*
then	DH1 EH EH NN1
thread	TH RR1 EH EH PA2 DD1
threaded	TH RR1 EH EH PA2 DD2 IH PA2 DD1
threader	TH RR1 EH EH PA2 DD2 ER1
threaders	TH RR1 EH EH PA2 DD2 ER1 ZZ
threading	TH RR1 EH EH PA2 DD2 IH NG
threads	TH RR1 EH EH PA2 DD2 ZZ
time	TT2 AA AY MM
times	TT2 AA AY MM ZZ
uncle	AX NG PA3 KK3 EL
whale	WW EY EL
whaler	WW EY LL ER1
whalers	WW EY LL ER1 ZZ
whales	WW EY EL ZZ
whaling	WW EY LL TH NG
year	YY2 YR
yes	YY2 EH EH SS SS

APPENDIX B

PARTS SUPPLIER LISTING

The components listed in the parts lists in Chapters Four, Five, Six, Seven, and Eight may be purchased at the following locations; however, some suppliers may not have all the parts required, so you may want to call before you make the trip.

> NOTE: For the complete kits all required parts may also be purchased at the following locations, with the exception of the printed circuit board (PC board). You may purchase a PC board kit from Radio Shack and use the complete kit circuit designs, or you may purchase a PC board already designed from R.I.S.T., Inc.

Call these corporate headquarters for locations in your area:

DISTRIBUTOR	*PARTS SUPPLIED*
Lionex Corporation 1 North Avenue Burlington, MA 01801 (617) 272-9400	SPO256-AL2, TTL logic, 7805 regulator, LM386 op amp
Pacesetter Electronics 3137 W. Warner Avenue Santa Ana, CA 92704 (714) 557-7131	SPO256-AL2, TTL logic, 7805 regulator, LM386 op amp
Future-Montreal 237 Hymus Boulevard Pt. Claire, Quebec Canada H95 5C7 (514) 694-7710	SPO256-AL2, TTL logic, 7805 regulator, LM386 op amp

DISTRIBUTOR	*PARTS SUPPLIED*
Anthen-Tustin 661 Dow Avenue Tustin, CA 92680 (714) 730-8000	SPO256-AL2, TTL logic, 7805 regulator, LM386 op amp
Arrow Electronics 25 Hub Drive Melville, NY 11747 (516) 391-1300	SPO256-AL2, TTL logic, 7805 regulator, LM386 op amp
Radio Shack a division of Tandy Corp. Fort Worth, Texas 76102	Breadboards, TTL logic, 7805 regulator, LM386 op amp, resistors, capacitors, jacks, potentiometers, PC board kits, cabinets, cable, connectors, switches

To purchase by mail, call or write to:

DISTRIBUTOR	*PARTS SUPPLIED*
R.I.S.T., Inc. Dept. M PO Box 499 Fort Hamilton Station Brooklyn, NY 11209 (212) 259-4934	*Speech Synthesizer Starter Kit* includes the SPO256-AL2, 3.12-MS2 ceramic resonator and the transistor transistor logic (TTL) parts ($14.95–$19.95). *Speech Synthesizer Complete Kit* is an unassembled module. Comes complete with the housing, PC board, synthesizer chip, and all the parts necessary for the project. The Exclusive Phrase Finder Software Program is also included on a cassette tape ($29.95–$49.95). *Speech Synthesizer Module* is completely assembled and ready for use. Supplied with the Exclusive Phrase Finder Software Program on a cassette tape and a 30-day limited war-

DISTRIBUTOR	*PARTS SUPPLIED*
	ranty on parts and workmanship ($39.95–$59.95).

NOTE: All parts listed above are available separately.

	Text-to-Speech Software allows you to enter in a text string composed of normal English words. Using a dedicated set of rules, the text string is converted to the appropriate allophone sounds by your computer. With this software program, your computer can speak anything that is typed into it or that is displayed on the screen. It makes vocabulary development for your own programs easy and efficient ($14.95–$24.95).

INDEX

Computers

HOW TO MAKE YOUR COMPUTER TALK

STEVEN J. VELTRI

Talking computers? This book tells you how to make it happen for under $35 to your Apple II, IIplus, and IIe; TRS-80 I, III, and IV; Commodore 64 and VIC 20; Sinclair ZX80 and ZX81; and Timex/Sinclair 1000 computers.

How To Make Your Computer Talk describes how complex speech waveforms are electronically modeled, enabling designers to put a digital model of the human vocal tract, a controller, and speech data on silicon. It provides you with complete, detailed instructions on interfacing speech hardware to the above mentioned machines.

Explore this exciting new dimension in computing and invent on your own, with these expert instructions.

Examples of software that illustrate fundamental principles are included as an introduction to speech synthesis using your home computer.

Steven J. Veltri is vice president of *Research in Speech Technology* and was formerly a product marketing engineer for the speech synthesis product lines of General Instruments Corporation.

McGraw Hill

Cover Design:
Lembit Rauk/Visible Ink

ISBN 0-07-067380-2